Raport Proiect TIA

Algoritm pentru prezicerea cotelor

de piață ale fotbaliștilor

Voicu Mihnea-Stefan 332AA

Cuprins

[**Introducere** 2](#_Toc185543650)

[Motivația și descrierea temei 2](#_Toc185543651)

[Tip de algoritm si algoritmul ales 2](#_Toc185543652)

[**Metodologie** 2](#_Toc185543653)

[**Descriere implementare** 3](#_Toc185543654)

[Particularități ale algoritmilor 3](#_Toc185543655)

[**Rezultate obținute** 5](#_Toc185543656)

[**Concluzii** 10](#_Toc185543657)

# **Introducere**

## Motivația și descrierea temei

Pentru acest proiect, mi-am dorit ca tema să fie în jurul unei pasiuni de ale mele, pentru a obține un set de date cât mai original. Din acest motiv, am ales să antrenez un algoritm să prezică cotele de piață ale jucătorilor de fotbal. Astfel, algoritmul o să primească cateva atribute ale jucătorului respectiv, cu care algoritmul o să fie antrenat ca apoi să prezică valorile dorite. Pentru a reuși sa producă aceste date, i-am dat algoritmului atribute din cel mai realist joc pentru simularea fotbalului, Football Manager.

## Tip de algoritm si algoritmul ales

Pentru acest tip de problemă, a fost necesar alegerea unor algoritmi de regresie. Întrucat tema nu este cea mai usoară, deoarece valorile respective care rezultă dintr-o combinație ale atributelor jucătorilor pot fi obținute in mai multe moduri, o să folosesc algoritmi de regresie neliniară, pentru a testa care se descurcă cel mai bine pe această temă.

Algoritmii de regresie neliniară pe care i-am ales pentru această testare sunt XGB Regressor, CatBoost Regression și Light GBM Regressor.

# **Metodologie**

## **Descriere implementare**

Pentru implementarea setului de date, mai întâi trebuie să observăm atributele care fac diferența în stabilirea cotei de piață a jucătorilor. Primul este “Position”, iar motivul pentru care face parte este că poziția jucătorilor determină o diferență, deoarece fundașii de regulă sunt mai ieftini, în timp ce atacanții dețin o valoare mai mare.

Cel de-al doilea atribut este “Ability”. Acesta reprezintă abilitatea fiecărui jucător în funcție de poziția în care joacă. Pentru a determina această valoare, se fac anumite calcule ale celor mai importante atribute tehnice, mentale și fizice pe care jucătorul respectiv le posedă.

Al treilea atribut este “Age”. Vârsta joacă un factor important un stabilirea valorilor de pe piață, deoarece jucătorii tineri care par a avea potențial sunt cei mai doriți. Astfel, prețul lor crește datorită cererii.

Ultimul atribut este “Reputation”. Reputația jucătorilor reprezintă cât de cunoscuți sunt aceștia de cele mai importante cluburi din lume. Chiar dacă un jucător este foarte bun și tânar, dar nu este la fel de cunoscut, acesta ar avea o cotă mai mică decât cineva de aceeasi abilitate, dar care este stiut la nivel mondial.

În final, toate aceste atribute sunt folosite pentru a determina “Transfer Value”-ul jucătorilor. Pentru acest algoritm, am oferit sume de genul 50, care reprezintă 50 de milioane. Am folosit această scriere pentru a ușura rezultatele obținute la Mean Square Error.

Pentru a realiza învățarea algoritmului, mai întâi am împărțit setul de date în două variabile, features, cu atributele fiecărui jucător, și target, pentru a determina obiectivul algoritmului. Apoi, am împărțit setul de date in 70 de exemple pentru antrenare și 20 de atribute pentru validare, pe care le-am folosit pentru a verifica corectitudinea algoritmului.

După ce i-am dat algoritmului să prezică, am calculat valorile MSE și R2, pentru a observa cât de corect a prezis și pentru a-l compara cu alți algoritmi. De asemenea, pentru rezultate am afișat și matricea de confuzie, necesară pentru a observa performanta algoritmului.

## Particularități ale algoritmilor

* XGB Regressor

Primul algoritm pentru regresia neliniara pe care l-am utilizat a fost acesta. Funcționarea acestuia se bazează pe mai multe seturi de algoritmi de gradient boosting, care au la bază arbori de decizie. Acești arbori au rolul de a realiza predicțiile algoritmului, printr-o învățare repetitivă. De fiecare dată când acești arbori sunt parcurși, li se adaugă un nivel nou care are rolul de a corecta greșelile făcute anterior.

* CatBooster Regressor

Acesta este cel de-al doilea algoritm de regresie neliniară. La bază, acesta funcționează tot prin intermediul unor arbori de decizie, care reprezintă tot e tehnică de gradient boosting, dar într-un mod diferit. În loc să corecteze greșelile făcute, fiecare arbore are rolul de a elimina erorile anterioare în momentul în care se adaugă un nivel nou.

* LightGBM Regressor

Acest algoritm este ultimul pe care l-am folosit de regresie neliniară. A fost dezvoltat la fel ca ceilalți, tot pe baza arborilor de decizie, însă ca și CatBooster Regressor, funcționează diferit. Arborele acestuia se dezvoltă prin a diviza doar o frunză, în funcție de rezultatul cel mai convenabil, această metodă numindu-se “leaf-wise”. Pentru eficientizarea algoritmului, acesta folosește o metodă numită Gradient-based One Side Sampling, care ajută la eliminarea punctelor cu gradiente mici.

# **Rezultate obținute**

* XGB Regressor

O imagine care conține text, captură de ecran, meniu, Font

Descriere generată automatDupă antrenarea algoritmului cu un set de date de lungime 70, i-am dat sa prezică valoarea a 20 de jucători. În următoarea poză, se pot observa index-ul jucătorului a cărui valoare a fost prezisă, “Transfer Value” reprezintă valoarea lui actuală, iar “Predicted Value” este valoarea prezisă de algoritm.

Rezultatele obținute arată că algoritmul prezice destul de bine valorile jucătorilor, cu o eroare de câteva milioane, care nu este prea mare, în cazurile cele mai bune, întrucât nu mă pot aștepta sa prezică exact valorile, iar aceste erori sunt de așteptat. MSE-ul are o valoare decentă ținând cont de numerele mari utilizatem, iar R2 arată o performanță de prezicere aproape foarte bună, in 67% din cazuri.

O imagine care conține text, captură de ecran, software, Dreptunghi

Descriere generată automatLa testare, rezultatele sunt asemănătoare, cu un MSE de 189 și un R2 de 0.45. Ținând cont de dimensiunea mică a setului de testare și de dificultatea, valoarea mică a lui R2 nu reprezintă un factor de îngrijorare.

Matricea de confuzie pentru testare este de 3 pe 3, deoarece a fost împărțită in 3 intervale ale valorilor prezise de algoritm. În acest caz, putem observa că algoritmul arată o performanță foarte bună, întrucât în majoritatea cazurilor a prezis exact, sau în intervalul adiacent.

Fiecare clasă o să aibă câte un set de indicatori de performanță, care indică cât de bine a performat algoritmul. În această situație, pentru primul interval (20,40) avem:

TP = 3 (elemental de pe linia 1 coloana 1)

FN = 3 (suma elementelor rămase pe linia 1)

FP = 0 (suma elementelor rămase pe coloanal 1)

TN = 9 (suma elementelor rămase în matrice)

Din acestea putem extrage că acuratetea este egală cu 0.8, iar precizia este chiar 1, cu toate că această valoare este improbabilă pentru un număr mai mare de date.

* CatBooster Regressor

O imagine care conține text, captură de ecran, meniu, Font

Descriere generată automatRezultatele acestui algoritm sunt aproape asemănătoare cu cele al primului, însă puțin mai slabe.

Putem observa totuși că acest algoritm a reușit să prezică mai cu exactitate valorile, atunci când le-a prezis corect, însă la cele pe care le-a prezis greșit, diferența a fost mai mare.

În ceea ce constă rezultatele setului de antrenare, modelul a reușit să performeze puțin mai bine, dar poate fi doar din cauza numărului mic de jucători. De asemenea, nu se observă nicio diferență in rezultatul matricii de confuzie, ceea ce poate însemna ca algoritmul s-a descurcat în parametrii optimi comparativ cu O imagine care conține text, captură de ecran, software, Software multimedia

Descriere generată automatprimul. Astfel, indicatorii de performanță ai acestuia sunt identici cu ai primului.

* O imagine care conține text, captură de ecran

  Descriere generată automatLightGBM Regressor

O imagine care conține text, captură de ecran, Dreptunghi, software

Descriere generată automatPerformanțele acestui algoritm sunt puțin mai bune decât celui de-al doilea, dar nu le depașesc pe ale primului, cu toate ca diferența este minusculă. Acest algoritm nu a reușit să prezică la fel de exact date ca cel precedent, însă a făcut mai puține erori mari și a avut multe predicții apropiate de valoarea reală.

Din punctul de vedere al datelor de testare, acest algoritm a avut cea mai bună performanță. Acesta a reușit să aibă cele mai bune valori pentru MSE și R2, iar matricea de confuzie indică faptul că algoritmul a reușit să prezică și mai bine valorile în intervalele respective. Pentru aceasta, indicatorii de performanță sunt (pentru clasa (20,40)):

TP = 5

FP = 1

FN = 1

TN = 8

Acuratețea o să rezulte 0.86, ceea ce indică o performanță mai bună față de primii 2 algoritmi, iar precizia este de 0.83, care este tot un rezultat foarte bun.

# **Concluzii**

În urma rezultatelor obținute si comparate între cei 3 algoritmi, pot spune că s-au efectuat preziceri la un nivel dorit, ținând cont de nivelul algoritmilor și de dimensiunea seturilor de date. Pentru o performanță mai bună, poate ar fi fost necesare mai multe atribute precum “Match Ratings” sau “Club Level”, care și ele la rândul lor pot afecta valoarea unui jucător, dar în același timp ar fi putut complica mai mult decât este nevoie algoritmul.

De asemenea, toți algoritmii au performat la un nivel asemănător, fiecare având anumite caracteristici proprii care îl diferențiau de ceilalți. De aceea, aș putea spune că cel care mi-a atras atenția cel mai mult a fost LightGBM Reggresor, deoarece, in opinia mea a performat cel mai constant la validare, iar la testare a oferit cele mai bune rezultate ale matricii de confuzie si indicatorilor de performanță.