

Program pentru clasificarea echipei castigatoare a unui joc MOBA

Ossian Mihai

29 mai 2024

1 Introducere

Un joc de tipul MOBA este bazat pe existența a 2 echipe, fiecare formate a câte 5 jucători. Scopul jocului este de a distruge baza inamică. Pentru a câștiga un meci jucătorii își vor folosi aptitudinile dar și unele mecanici introduse de către creatorii jocului pentru a-și ridica șansa de victorie.

În acest proiect voi lua ca exemplu de joc MOBA League of Legends pentru a avea un set de mecanici cunoscute deja pentru a ușura prezicerea echipei câștigătoare. În LoL obiectivul principal este Nexus-ul inamic, prin distrugerea acestuia jocul se va termina iar echipa care reușește aceasta va fi numită câștigătoare. Ca mecanici adăugate de către dezvoltatori avem un număr destul de bogat: Iteme (lucruri care fac caracterele mai puternice), obiective (set de NPC care au un impact asupra puterii caracterelor), abilități etc. Toate acestea au un impact mai mare sau mai mic asupra echipei câștigătoare, dar desigur, performanța jucătorilor unul din cel mai mare impact a decizie finale de câștig.

CAND FOLOSIM REFERINTE DIN ALTE ARTICOLE LE NUMEROTAM, 1 2 3 ETC SI LA BIBLIOGRAFIE Scriem 1. LINK + W/E

2 Baza de date

Baza de date este concepută din 50.000 de jocuri din anul 2019 (<https://www.kaggle.com/datasets/datasnaek/league-of-legends/data>). Jocurile sunt luate din modul de joc "Ranked Solo/Duo" adică competențele și aptitudinile jucătorilor sunt aflate aproximativ la același nivel.

2.1 Descrierea bazei de date

O mare parte a bazei de date este exprimată sub forma unui rezultat binar (1 sau 2 pentru echipa 1 sau echipa 2). Baza aleasă este concepută din următoarele:

- gameDuration - Durata jocului exprimată în secunde
- winner - Care echipă a câștigat jocul
- firstBlood - Echipa care făcut primul asasinat
- firstTower - Echipa care a distrus primul turn
- firstInhibitor - Echipa care a distrus primul inhibitor
- firstBaron - Echipa care a ucis primul Baron Nashor (Baron)
- firstDragon - Echipa care a ucis primul Dragon
- firstRiftHerald - Echipa care a ucis primul Rift Herald
- t1/t2_towerKills - Câte turnuri au fost distruse de către fiecare echipă
- t1/t2_inhibitorKills - Câte inhibitoare au fost distruse de către fiecare echipă
- t1/t2_baronKills - Câți Baron au fost uciși de către fiecare echipă
- t1/t2_dragonKills - Câți Dragon au fost uciși de către fiecare echipă
- t1/t2_champ1-5id - Caracterul ales de către fiecare jucători în joc, exprimat în ID.

2.2 Ce dorim să obținem

Obiectivul principal a cercetării este de a determina (clasifica) automat care echipă este cea câștigătoare.

Clasificarea se poate folosi în viitor pentru o aplicație care primește date în timp real și rezultă clasificarea echipei care are un avantaj și poate fi câștigătoare. Aceasta poate crea un indicator vizual a echipei în avantaj pentru scena E-Sports pentru spectatori.

Aceste informații pot fi afișate și sub forma unui tutorial pentru jucătorii începători, în timp real. De exemplu: Pentru a avea o șansă ridicată pentru victorie încearcă să distrugi 3 turnuri.

3 Stagiul actual a cercetării

Având în vedere creșterea scenei E-Sports din ultimii 10 ani, această piață este foarte tânără dar în rapidă extindere, plină de sponsori formați din corporații de nivel global, de exemplu Mercedes-Benz, Louis Vuitton, Red Bull, Spotify și mulți alții.

În 2015 Thomas Huang, David Kim și Gregory Leung creează o unealtă care primește numele celor 10 jucători și caracterul ales de către fiecare jucător. Aceasta va afișa care este rata de câștig a echipei, decizie luată în funcție doar de jocurile trecute din istoricul jucătorilor. —1—

În 2023 scena E-Sports a observat o creștere extrem de mare în numărul mediu a spectatorilor în timp real de la 0.67 milioane la 1.26 milioane, cu cel mai mare număr de vizualizări de la 1.62 milioane în 2016 la 6.4 milioane în 2023. Aceste numere au stârnit nevoia de a ajuta spectatorii să observe avantajul foarte subtil între cele 2 echipe a cărei diferență de experiență este mică, aproape indiscutabilă. În tot același an Adrian Ramdo Firmansyah creează un predictor al câștigurilor care folosește ca date suport informații despre joc din primele 10 minute și creează un procentaj de victorie al celor 2 echipe. —2—

Tot în această perioadă Riot Games anunță un parteneriat cu AWS, aceștia revizuiesc modalitatea de a observa avantajele echipei de pe parcursul jocului, lucru care ușurează munca comentatorilor live și stârnește curiozitatea spectatorilor cu informații pe care aceștia nu le pot vizualiza live. —3—

Pe lângă scena E-Sports, un utilizator GitHub sub numele de Giantpizzahead concepe un program care primește date live din jocul curent și afișează șansa de câștig a jucătorului cu aplicația instalată. Acuratețea acestuia este în jurul ratei de 70%. —4—

4 Modele de învățare automată

Pentru predictorul nostru v-am observa o multitudine de modele de învățare automată și eficiența acestora. Algoritmii de învățare automată aleși sunt următorii: Decision Tree, Deep Learning, Naive Bayes (Kernel) și Gradient Boosted Tree. Pentru aplicarea acestora am folosit RapidMiner Studio Educational.

Deep Learning este modelul care prezintă cel mai bun scor cu 50114 de cazuri clasificate corect și doar 1376 greșite, acuratețea fiind de 97.33%.

accuracy: 97.45%

	true range1	true range2	class precision
pred. range1	25622	858	96.76%
pred. range2	455	24555	98.18%
class recall	98.26%	96.62%	

Figura 1: Matricea de acuratețe Deep Learning

Următoarele 2 modele luate în ordinea acurateței sunt Gradient Boosted Tree și Decision Tree, acestea având rezultate foarte similare cu acuratețe de 96.77% respectiv 96.74%.

accuracy: 96.77%

	true range1	true range2	class precision
pred. range1	25528	1114	95.82%
pred. range2	549	24299	97.79%
class recall	97.89%	95.62%	

Figura 2: Matricea de acuratețe Gradient Boosted Tree

accuracy: 96.74%

	true range1	true range2	class precision
pred. range1	25489	1089	95.90%
pred. range2	588	24324	97.64%
class recall	97.75%	95.71%	

Figura 3: Matricea de acuratețe Decision Tree

În cele din urmă avem Naive Bayes care prezintă cea mai slabă performanță dintre cele 4 modele alese. Acuratețea acesteia este de 93.46%.

accuracy: 93.46%

	true range1	true range2	class precision
pred. range1	24605	1896	92.85%
pred. range2	1472	23517	94.11%
class recall	94.36%	92.54%	

Figura 4: Matricea de acuratețe Naive Bayes

5 Implementarea modelului Deep Learning

Dintre cei 4 algoritmi de învățare automată am ales Deep Learning pentru acuratețea excepțională a clasificării.

Matricea de corelare ne ajuta să observăm importanța și influența datelor asupra șansei de câștig. Aceasta arata astfel.

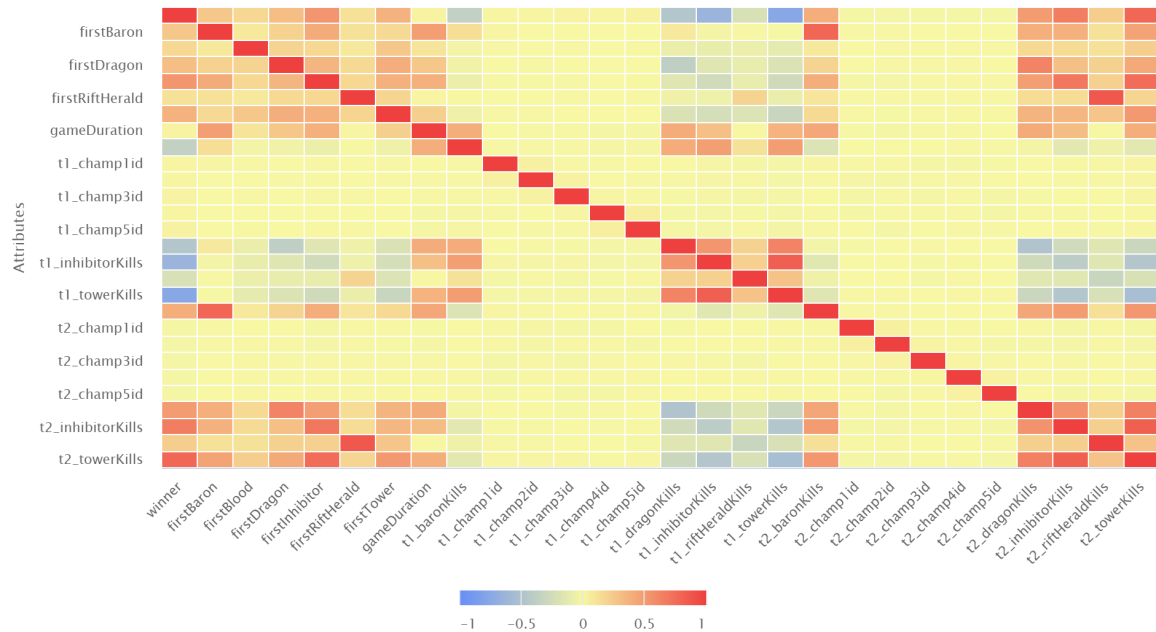


Figura 5: Matricea de corelare

După cum putem observa o mare parte din datele noastre sunt irelevante pentru decizia finală a echipei câștigătoare. Următoarele date au o relevanță ridicată:

Attributes	winner ↑
t1_towerKills	-0.772
t1_inhibitorKills	-0.649
t1_dragonKills	-0.472
t1_baronKills	-0.369
t1_riftHeraldKills	-0.218

t2_towerKills	0.786
t2_inhibitorKills	0.660
firstInhibitor	0.536
t2_dragonKills	0.497
t2_baronKills	0.399
firstTower	0.376
firstDragon	0.310
firstBaron	0.262
t2_riftHeraldKills	0.226
firstBlood	0.175
firstRiftHerald	0.120

Datele care prezintă o importanță mai scăzută sunt legate despre campionul pe care fiecare jucător îl alege și durata jocului.

Pentru calcularea gini indexului am folosit Python iar acesta arată în felul următor:

```
winner 0.5000831496411784
firstBaron 0.33930710015362364
firstBlood 0.48970980493094485
firstDragon 0.46342289725398506
firstInhibitor 0.40054791829618375
firstRiftHerald 0.37938551914267704
firstTower 0.47766828035137476
gameDuration 0.000678385019787682
t1_baronKills 0.5371915843577076
t1_champ1id 0.010175721736569512
t1_champ2id 0.010539146862985838
t1_champ3id 0.010486902992324175
t1_champ4id 0.010526145301930986
t1_champ5id 0.010239212504461625
t1_dragonKills 0.24155876609414662
t1_inhibitorKills 0.32485403456318096
t1_riftHeraldKills 0.6235379959589172
t1_towerKills 0.08919695124520215
t2_baronKills 0.5091493950275952
t2_champ1id 0.010257348305408517
t2_champ2id 0.010479934125423989
t2_champ3id 0.010342692407921922
t2_champ4id 0.01049192558550684
t2_champ5id 0.010300881469629774
t2_dragonKills 0.23903032418922898
t2_inhibitorKills 0.3358629349558718
t2_riftHeraldKills 0.6350909522751431
t2_towerKills 0.09087637499429037
```

Figura 6: Gini Index

6 Testarea modelului folosind noi date

Pentru testarea modelului vom folosi simulatorul creat în Rapid Miner. Datele de intrare sunt preluate din meciuri deja existente și v-om aprecia corectitudinea clasificării. De asemenea id-ul caracterelor îl v-om lua ca irelevant, deoarece acesta nu pare să schimbe răspunsul final.

După cum putem observa, certitudinea simulatorului pentru datele introduse este mai mare de 90%, iar clasificarea fiind corectă în 5/5 exemple.

Joc# ->	1	2	3	4	5
firstBaron	2	1	0	1	2
firstBlood	2	2	2	2	1
firstDragon	2	1	2	2	2
firstInhibitor	2	1	1	2	1
firstRiftHerald	2	1	1	1	2
firstTower	2	1	1	2	2
t1BaronKill	0	3	0	1	1
t1DragonKill	0	4	1	2	2
t1InhibitorKill	0	7	1	0	3
t1RiftHeraldKill	0	1	1	1	0
t1TowerKill	0	11	9	7	11
t2BaronKill	1	0	0	0	1
t2DragonKill	2	1	1	3	3
t2RiftHeraldKill	1	0	0	0	1
t2TowerKill	10	2	0	10	4
t2InhibitorKill	2	0	0	3	1
ClasificareSimulator	2	1	1	2	1
Câștigător	2	1	1	2	1
Certitudine	97.45%	89.23%	98.34%	99.92%	100.00%

Figura 7: Rezultate

7 Concluzii

După câte putem observa, simulatorul nostru prezintă o clasificare de acuratețe excepțională pentru echipele de un nivel asemănător, astfel obiectivul nostru principal este atins.

De asemenea, am reușit să realizăm un produs care nu este prezent în piață cu aceleași descrieri.

8 Bibliografie

Sursele informațiilor folosite:

—1— <https://thomasyluang.github.io/League-Predictor/>

—2— <https://medium.com/@adrianfirmansyah33/league-of-legends-winner-prediction-using-machine-learning-664cc062f646>

—3— <https://lolesports.com/article/dev-diary-win-probability-powered-by-aws-at-worlds/blt403ee07f98e2e0fc>

—4— <https://github.com/Giantpizzahead/league-win-predictor>

Baza de date folosită - <https://www.kaggle.com/datasets/datasnaek/league-of-legends/data>