Proiect pclp3

Construirea dataset-ului

Am pornit de la un dataset de pe Kaggle (<https://www.kaggle.com/datasets/ulrikthygepedersen/online-chess-games>) care inregistreaza 20058 meciuri jucate pe platforma online de sah Lichess.

Informatiile initiale ale dataset-ului sunt : id meciului, daca a fost sau nu un meci competitiv(rated), numarul de mutari, victory\_status(conditia prin care s-a incheiat meciul), winner(daca a existat,altfel valoarea ei este Draw), time-increment care are doua componente(x+y) unde x reprezinta timpul initial al fiecarui jucator in minute,iar y reprezinta in secunde timpul castigat de un jucator dupa realizarea unei mutari in secunde, white/black\_id numele jucatorului, white/black rating elo-ul jucatorului, moves ce contine toate mutarile, opening\_moves ce reprezinta numarul de mutari considerate ca parte dintr-o deschidere clasica, opening\_name/short\_name numele lung si scurt al deschiderii jucate.

La acesta am adaugat ulterior 3 noi coloane care reprezinta varsta,sexul si nationalitatea jucatorului care a castigat partida inregistrata(in cazul unei egalitati voi considera pentru aceste campuri jucatorul cu un avantaj de material). Datele suplimentare pentru aceste 3 coloane au fost generate in mod aleatoriu, folosind atat functii de generare a numerelor aleatorii,cat si libraria Faker pentru a genera nationalitatile jucatorilor. De asemenea am eliminat coloanele care ar complica prea mult modelul,dar si pe cele care nu ajuta in predictiile dorite, acestea fiind 'opening\_fullname', 'opening\_variation', 'opening\_moves', 'opening\_response', 'white\_id', 'black\_id', 'game\_id';

Tipul problemei

Dorim pe baza acestuia dataset sa prezicem ce jucator va castiga,dar si in ce conditii va castiga acesta, asadar aceasta este o problema de tip clasificare dubla(clasificam atat dupa castigator,dar si dupa modalitatea de a castiga)

Prelucrarea dataset-ului

Pe langa eliminarea coloanelor redundante, am spart campul time\_increment in 2, timp initial si incrementarea in secunde, avem astfel 2 campuri numerice noi. Fiind o problema de regresie am mapat coloanal de winner la 3 valori numerice (Alb = 1, Negru = 2, Egalitate = 0), iar conditia de finalizare a meciului in 4 valori numerice (Mat = 0, Capitulare = 1, Lipsa de timp = 2, Egalitate = 3)

Analiza informatiilor initiale(inainte de procesari)

Folosesc functia describe pentru a analiza informatiile, observam urmatoarele:

(am analizat direct dataset-ul mare, pentru a incerca sa intuim aproximarile finale)

observam ca numarul mediu de mutari este 60, avem totusi si meciuri aberant de lungi(maxim 349)

media pentru statutul de victory este 0.85 deci tendinta este ca un meci sa se termine fie in mat sau capitulare

rating-ul pentru jucatori este in jur de 1590, exista o disperie mare(jucatori intre 784-2700)

media timpului(minute) este de 13 minute

acesta este incrementat dupa fiecare mutare cu in medie 5.2 secunde

varsta medie a jucatorilor este de 38 de ani cu interval intre 14 si 64 de ani

tendinta generala arata ca meciurile se termina intr-o victorie a jucatorului alt ca rezultat al mediei 1.4

alb - 1, negru - 2, egal – 0

Analiza exploratorie a datelor

A group of blue and white graphs

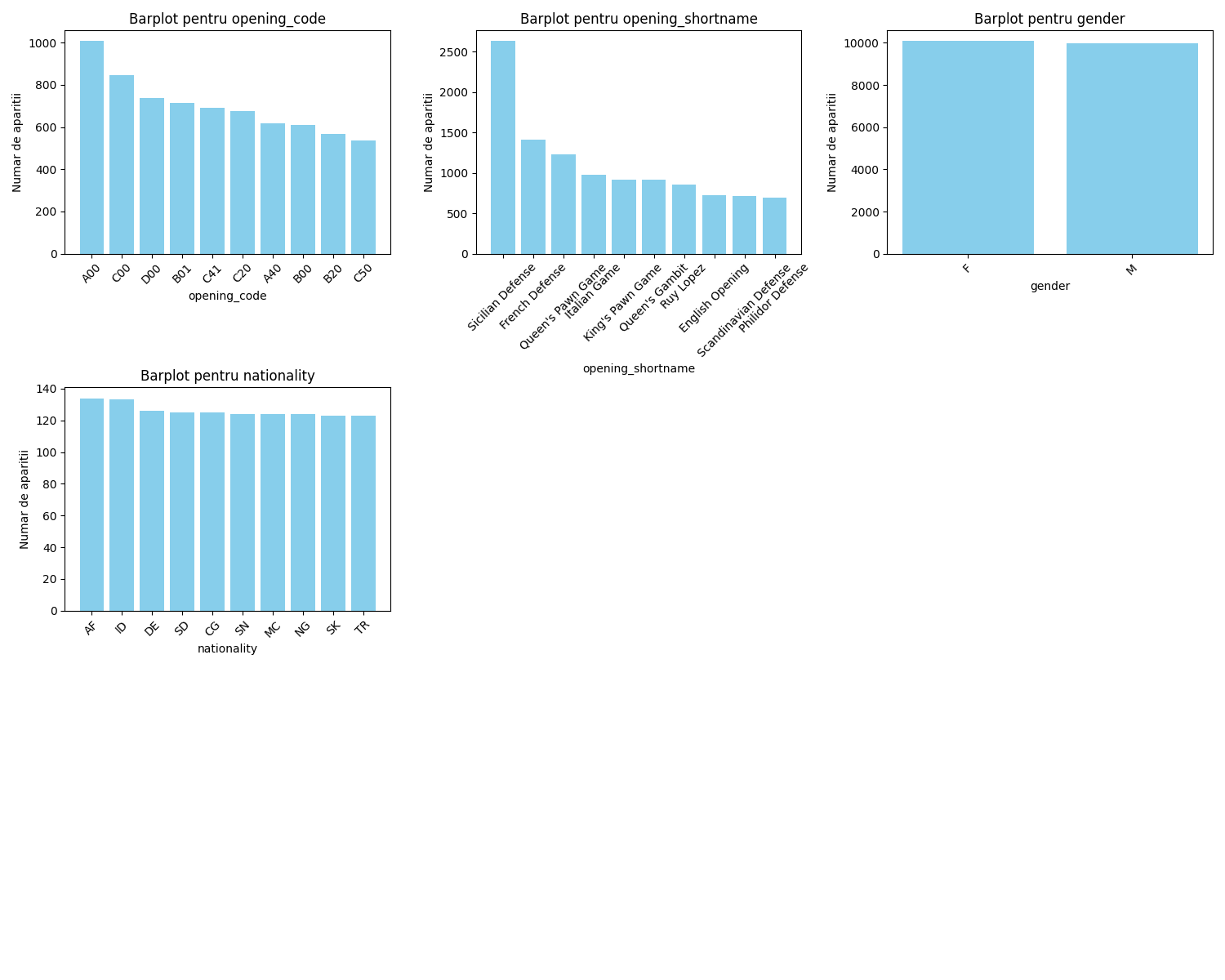
AI-generated content may be incorrect.

Histrograma pentru variabilele numerice

Observam ca tendinta este ca meciurile sa se termine in jur de 50-80 de mutari, rating-ul mediu al jucatorilor este de 1500(in mod evident este similar si pentru alb si pentru negru,conform asteptarilor),din initial\_time in jur de 5 minute putem determina ca majoritatea meciurilor sunt de tip blitz), din histograma pentru castigator observam o tendinta usoara ca meciurile sa se castigate de alb, spre deosebire de negru, observam si ca in dataset-ul primit, egalitatea este o raritate(probabil datorita ratei mari de capitulare in cazul unui meci pierdut,care in schimb are sanse sa se transforme intr-un egal), lucru confirmat si de histograma pentru conditia de victorie, meciurile terminate prin capitulare fiind cele mai comune , urmate de cele terminate prin mat.

Deoarece varstele sunt generate aleatoriu, informatia din tabelul de varsta in practica este irelevanta, lucru relativ evident din grafic.

Barplot pentru variabilele categorice



Aici ne intereseaza de fapt doar primele 2 barplot-uri deoarece restul sunt generate aleatoriu de catre noi,

Observam cu usurinta popularitatea Defensivei Siciliene si celei Franceze cand vine vorba de deschideri clasice, lucru care se reflecta si in coduri de deschidere A00, C00 si D00 sunt asociate printre altele si cu cele mai populare deschideri din baza noastra de date.

Matricea de corelatii

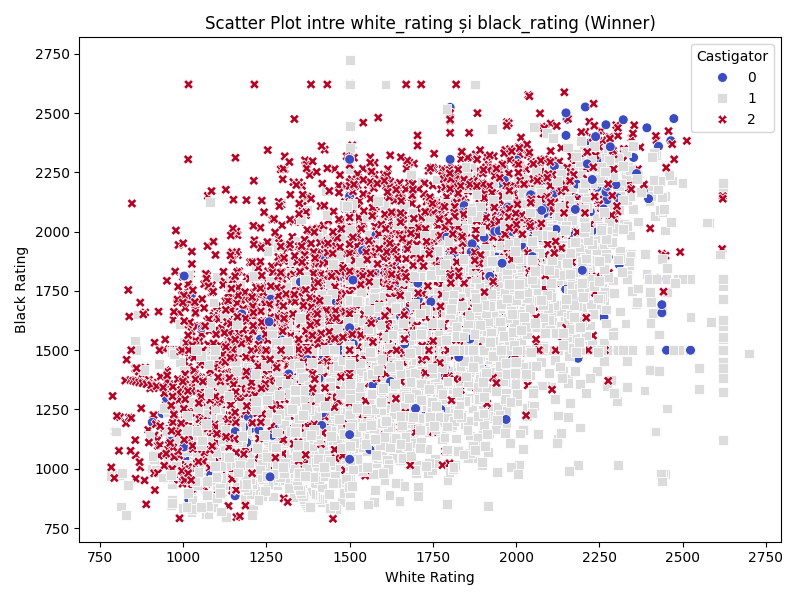
A blue and red squares with white text

AI-generated content may be incorrect.

Din analiza heatmap-ului observam ca nu exista o legatura intre varsta si castigator, deoarece varsta este generata aleatoriu in model, exista o legatura liniara moderata, de unde reiese faptul ca in general

jucatorii sunt plasati in meciuri cu oponenti de un nivel similar (lucru pe care-l intuim cu usurinta), observam ca exista o legatura slaba intre cine castiga si modul de castig.

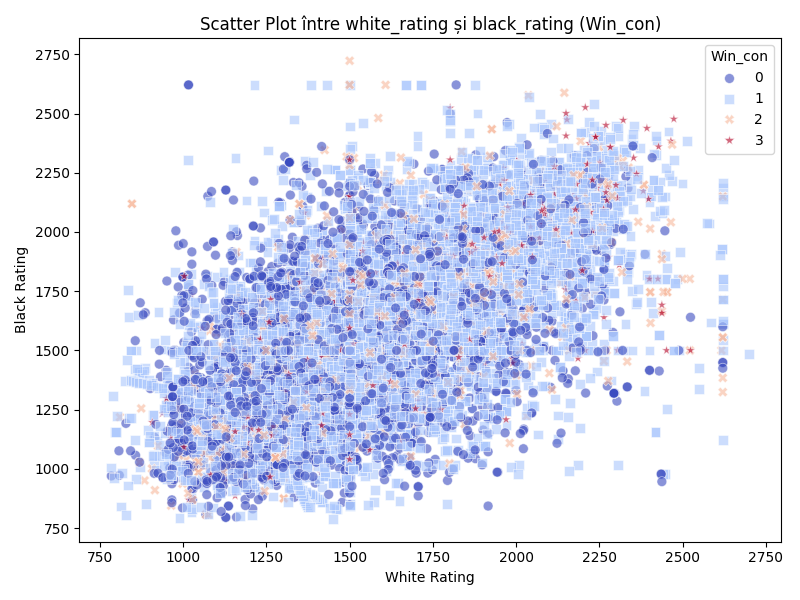
Winner vs rating



Observam ca tendinta generala este ca la nivel inalt negru si alb castiga cam la fel de mult,la egalitate cu sansa de a se termina in egal meciul, pentru nivelurile de joc sub 2000 rating, tendinta este ca meciurile sa se termina fie in win/lose, aproape niciodata in egal,

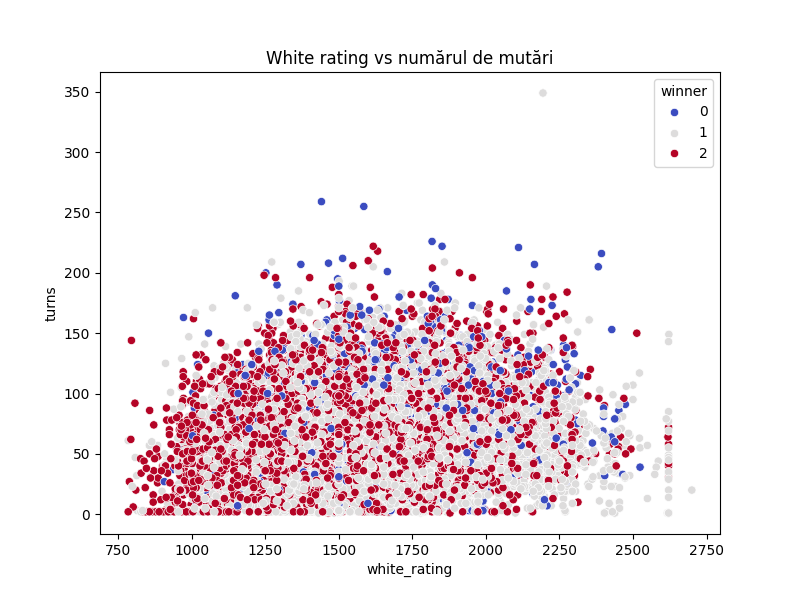
unde in continuare alb la nivel egal de rating are un avantaj.

Winner vs win\_con



Observam ca la nivel inalt majoritatea meciurilor se termina in resign sau egalitate,dar si pierderea prin lipsa de timp este frecventa,comparativ cu incheierea meciului prin mat, probabil ca rezultat al unei intelegeri mai profunde a jocului. In schimb pentru restul jucatorilor, majoritatea meciurilor se termina prin resign sau mat.

Rating vs turns



Observam ca odata cu cresterea rating-ului, creste si numarul de mutari facute de jucator, insa media pentru jucatorii de nivel inalt pare sa fie undeva la 120 – 140 de mutari.(aici se considera toate mutarile din joc, asadar un jucator are in medie 60-70 de mutari), intuim cu usurinta ca acest lucru se datoreaza atat experientei jucatorului analizat,dar si a oponentului acestuia.

Preprocesarea informatiilor

Inainte de a crea dataset-urile de antrenare si testare am realizat prelucrarile necesare pentru folosirea unui model de regresie. Pentru valorile categorice aplic one-hot encoding (rated, opening\_shortname, nationality, opening\_code), label-encoding pentru sex, iar pentru valorile numerice (turns, white\_rating, black\_rating, age, initial\_time, increment\_sec) voi standardiza valorile folosind StandardScaler.

Crearea dataset-urilor

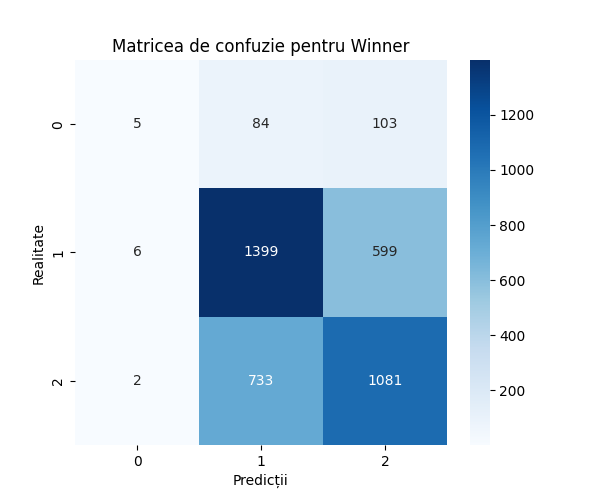
Dupa preprocesare folosim functia train\_test\_split si creez dataset-urile de testare si antrenare, impartim datele: 80% pentru antrenament, 20% pentru test apoi le salvez separate in csv-uri.

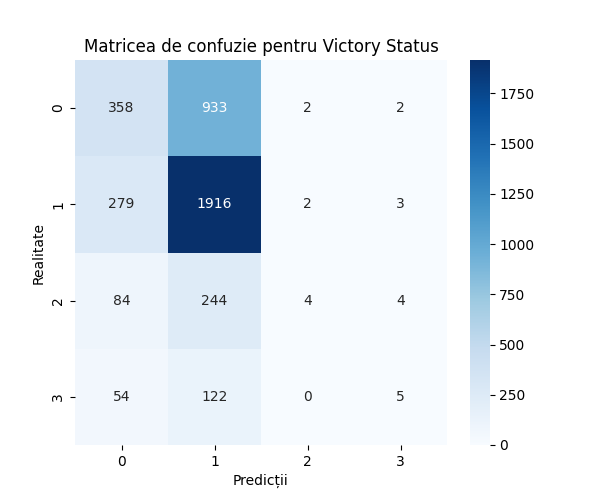
Antrenarea modelului

Vom folosi logistic regression impreuna cu MultiOutputClassifier(modalitate prin care pot returna mai multe clasificari ca output, spre deosebire de logistic regression care are doar un output).

Analiza acuratetii modelului

Deoarece nu pot folosi matricea de confuzie direct pe ambele rezultate returnate, vom face o matrice pentru fiecare componenta a predictiei returnate.





Rezultatele testelor de acuratete

Acuratețe winner: 0.6194

Acuratețe victory\_status: 0.5690