Wtęp

Model ma za zadanie ogadnąć który gracz wygra partię szachów. Oczywiście nie da się przewidzieć kto wygra, bo to zawsze zależy od wielu czynników, ale byłem ciekaw, czy kiedy zna się ELO graczy, zagrany debiut, długość fazy debiutowej oraz format gry, czy da się przewidzieć, jaką szansę na wygraną ma każdy z graczy? Byłem też ciekaw, jak ELO graczy wpływa na popularność debiutu, skuteczność debiutu, oraz status zakończenia (mat, rezygnacja, itd).

Baza danych

Bazy danych pobieram z Kaggle. Istnieją tam wielkie bazy, posiadające nawet ponad kilka milionów partii, jednak na początku użyłem plik z 20,000+ partiami. Nie musiałem za bardzo modyfikować tych danych, jednak usunąłem parę niepotrzebnych parametrów i parę parametrów które nie powinny być znane podczas fazy debiutowej (takie jak długość partii w turach). Zmodyfikowałem równiez White ELO i Black ELO na Average ELO oraz ELO difference, ponieważ dane w takim formacie bedą wydajniejsze i skuteczniejsze. Aby zwiększyć wydajność, dane zostały znormalizowane.

Klasyfikacja

Użyłem Decision Tree, K Nearest Neighbours, Naive Bayes, oraz sieci neuronowych. Spodziewałem się, ze sieci neuronowe okażą się najskuteczniejsze, jednak zaskakująco wysoki wynik osiągnął też Decision Tree.

Wyniki:

```
Decision Tree Accuracy: 0.5741110003323363
KNN Accuracy: 0.5807577268195414
Naive Bayes Accuracy: 0.13742107012296445
Neural Network Accuracy: 0.6191425722831505
```

Nie są to bardzo dokładne modele, jednak należy pamiętać, ze nie da się przewidzieć kto wygra w szachach. Jedyne co robimy to dajemy każdemu z graczy prawdopodobienstwo.

Patrząc na te wyniki z testów, postanowiłem odrzucić KNN oraz Naive Bayes. Decision Tree również tez musiałem odrzucić, ponieważ model ten nie pozwala na obliczania prawdopodobieństw. Zostały więc sieci neuronowe. Następnie, użyłem ten wytrenowany model na kilku moich partiach. zeby zobaczyć, jakie miałem szanse na zwycięstwo. Oto kilka wyników z gier z pliku new_game.csv:

```
rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,B02,6,1026,979,white
Neural Network Probabilities:
 black    0.306370
draw     0.044712
white    0.648918
Name: 0, dtype: float64


rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,B02,5,1012,987,white
Expected winner: white
Neural Network Probabilities:
 black    0.324784
draw     0.046840
white    0.628376
Name: 1, dtype: float64

rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,B02,4,1050,994,white
Expected winner: white
Neural Network Probabilities:
 black    0.296836
draw     0.045555
white    0.657610
Name: 2, dtype: float64

rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,D00,2,1201,1001,white
Expected winner: white
Neural Network Probabilities:
 black    0.137303
draw     0.067924
white    0.794774
Name: 3, dtype: float64
```

```
Name: 3, dtype: float64

rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,C44,6,991,965,white
Expected winner: white
Neural Network Probabilities:
 black    0.316355
draw     0.036637
white    0.647007
Name: 4, dtype: float64

rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,B27,3,949,972,white
Expected winner: white
Neural Network Probabilities:
 black    0.332836
draw     0.045496
white    0.621667
Name: 5, dtype: float64

rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,B22,6,979,976,white
Expected winner: white
Neural Network Probabilities:
 black    0.430961
draw     0.029385
white    0.539654
Name: 6, dtype: float64

rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,B02,5,1012,1300,black
Expected winner: black
Neural Network Probabilities:
 black    0.723016
draw     0.039079
white    0.237905
Name: 7, dtype: float64

rated,increment_code,opening_eco,opening_ply,white_rating,black_rating,winner
TRUE,10+0,B02,4,1050,1250,black
Expected winner: black
Neural Network Probabilities:
 black    0.640921
draw     0.043535
white    0.315544
Name: 8, dtype: float64
```

Jak widać, w każdym przykładzie model poprawnie odgadnął gracza który wygrał. W większości miejscach gracz wygrywający miał więcej ELO, więc dałoby się odgadnąć i bez modulu, jednak w partii 6 zwycięzca miał mniej ELO, co oznacza, ze model poprawnie wziął pod uwagę tez inne czynniki, takie jak debiut. Ostatnie 2 partie posiadają zmodyfikowe ELO aby zobaczyć, czy model poprawnie da większą szansę graczowi z większym ELO.

Asocjacje

Ostatnim etapem jest szukanie ciekawych zasad z użyciem apriori.

Podzieliłem ELO na dwie kategorie, HIGH, które oznacza bardzo wysokie ELO, jednak nie na poziomie arcymistrzów, a niższe na LOW

Na początku, zbadajmy jak ELO graczy ma znaczenie na status zakończenia gry:

```
   antecedents  consequents   support  confidence      lift
8       (high)     (resign)  0.050454    0.677830  1.219692
14       (low)     (resign)  0.505285    0.545920  0.982332
10       (low)       (mate)  0.304617    0.329114  1.043694
4       (high)       (mate)  0.010719    0.144005  0.456673
6       (high)  (outoftime)  0.007428    0.099799  1.191530
12       (low)  (outoftime)  0.076329    0.082467  0.984597
0       (high)       (draw)  0.005833    0.078366  1.734944
3        (low)       (draw)  0.039336    0.042499  0.940896
```

Patrząc na wyniki można odczytać, ze gracze z niskim ELO zdecydowanie częściej kończą grę matem. Wynika to z tego, ze doświadczeni gracze wiedzą, kiedy partia jest nie do wygrania, i rezygnują, by nie tracić swego czasu oraz przeciwnika, podczas gdy na niższych poziomach graczę częściej liczą na błąd przeciwnika.

Następnie debiuty. Spróbujemy zobaczyć, jak ELO wpływa na skuteczność openingu. Interesują więc nas asocjacje typu (ELO, OPENING) -> (WINNER)

Niestety, z powodu małej ilości danych, istnieje za mało danych o grach z wysokim ELO. Nie ma ani jednej zasady, które przekracza 0.001 pewności. Skupimy się więc na grach na niskich poziomach.

```
 231  (A13, low)     (white)  0.001695    0.755556  1.515342
441  (C70, low)     (white)  0.002293    0.696970  1.397842
374  (C34, low)     (white)  0.002393    0.657534  1.318750
460  (D06, low)     (white)  0.005285    0.642424  1.288446
456  (D04, low)     (black)  0.001296    0.634146  1.396696
```

Oznacza to więc, ze statystycznie, na podstawie posiadanych danych, jezeli chce się wygrać na niskich poziomach, powinno się grać Ruy Lopez jako biały, i Queen's Pawn Game jako czarny.

```
(low, C44)     (white)  0.009522    0.517615  1.038129
(B02, low)     (white)  0.004138    0.525316  1.053574
```

Dowidziałem się również, ze grając moim ulubionym debiutem jako białym (Ponziani Opening C44) mam odrobinę lepsze szanse, ale jako czarny (Alekhine's defence B02) mam gorsze szanse.

Byłem też ciekaw jak skuteczny jest Polish Opening dla białych na niskich poziomach, i niestety nie jest za skuteczny.

```
(A00, low)     (black)  0.027720    0.570842  1.257269
```

Ostatecznie, byłem ciekaw, jak skuteczne są debiuty na niskich poziomach, takie co są często spotykane na poziomie arcymistrzów, na przykład Sicilian Defence (B20), Queen's Gambit (D06 Accepted, D07 Denied), czy Ruy Lopez (C70)

```
 441  (C70, low)     (white)  0.002293    0.696970  1.397842
460  (low, D06)     (white)  0.005285    0.642424  1.288446
293  (B20, low)     (black)  0.015006    0.562617  1.239153
463  (low, D07)     (white)  0.001147    0.522727  1.048382
```

Jak widać, debiuty te są bardzo skuteczne nawet na niskich poziomach.

Podsumowanie

Udało się wytrenować model, które po fazie debiutowej potrafi powiedzieć, która strona ma większe szanse na zwycięstwo. Poprzez zasady asocjacyjne, dowiedziliśmy się, którymi debiutami powinno się grać na nizszych poziomach, zeby zmaksymalizować swoją szansę na zwycięstwo.

Źródła

https://www.365chess.com/eco.php ECO debiutów szachowych. https://www.kaggle.com/datasets/datasnaek/chess Dataset https://www.chess.com Przkłady gier do przetestowania.