**Kamil Tatrocki** **280506**

projekt z metod systemowych i decyzyjnych

Spis treści

[OPIS ZBIORU DANYCH 2](#_Toc198736401)

[EKSPLORACJA ZBIORU DANYCH 2](#_Toc198736402)

[MODELE MASZYNOWE BEZ OPTYMALIZACJI 7](#_Toc198736403)

[Optymalizacja modeli maszynowych 8](#_Toc198736404)

# OPIS ZBIORU DANYCH

W niniejszym projekcie analizuję dane pochodzące z gry FIFA 22, stanowiące część większego zbioru obejmującego statystyki zawodników z trybu Kariery w wersjach FIFA od 2015 do 2022 roku. Skupiłem się wyłącznie na danych z najnowszej dostępnej edycji – FIFA 22 – zawartych w pliku players\_22.csv. Dane te zawierają informacje o ponad 100 atrybutach zawodników, w tym statystyki związane z umiejętnościami ofensywnymi, defensywnymi, fizycznymi oraz mentalnymi. Uwzględnione są również dane personalne graczy, takie jak narodowość, klub, pozycja na boisku czy wynagrodzenie.

# EKSPLORACJA ZBIORU DANYCH

Obraz zawierający zrzut ekranu, origami

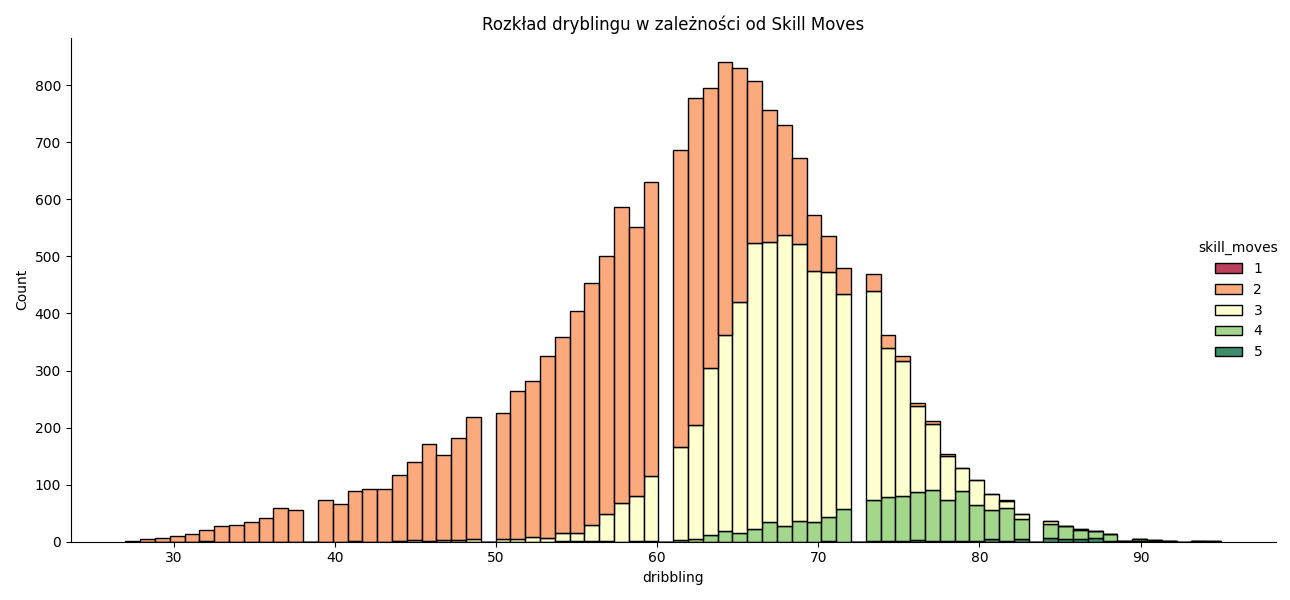
Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Wnioski:

- Na podstawie tego wykresu widać wyraźny wzrost mediany wraz z rosnącą reputacją międzynarodową, czyli te dwie dane są ze sobą skorelowane.

- Zawodnicy o reputacji międzynarodowej równej 1 mają zdecydowanie najszerszy zakres oceny ogólnej, a pozostałe wartości reputacji międzynarodowej mają bardziej zbliżony kształt.

- Dla reputacji międzynarodowej równej 4 oraz 5 zawodnicy mają zdecydowanie węższy zakres oceny ogólnej, co sugeruje, że Ci zawodnicy są bardziej zbliżeni do siebie.

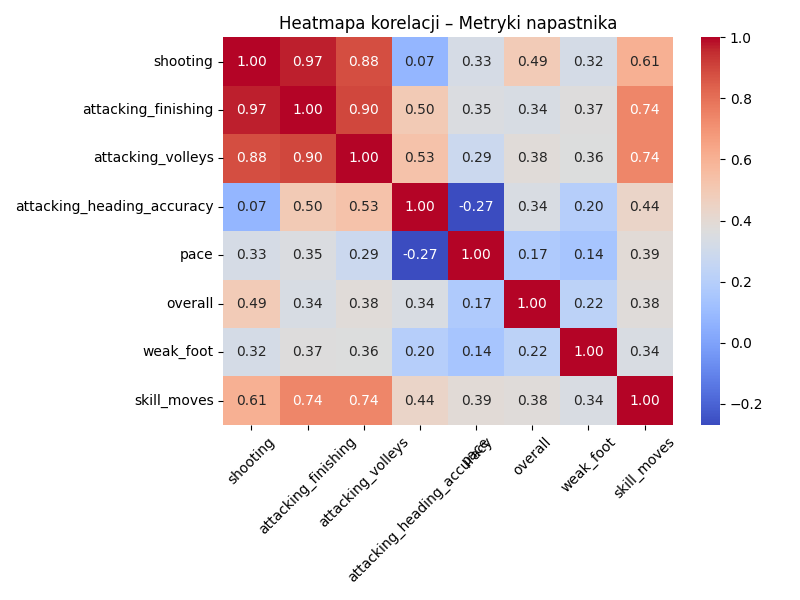


Wnioski:

- Z histogramu widać, że im wyższa wartość dryblingu tym większą ilość sztuczek ma zawodnik. Sugeruje to całkiem wysoką korelację między dryblingiem, a liczbą gwiazdek sztuczek.

- Liczba gwiazdek sztuczek nie jest zbalansowana. Zdecydowanie przeważa ilość 2 gwiazdek sztuczek. Ilość gwiazdek sztuczek równa 1 występuje bardzo rzadko. Po przefiltrowaniu zbioru danych zauważyłem, że ta ilość gwiazdek sztuczek występuje dla bramkarzy.

- Rozkład statystyki dryblingu jest całkiem zbalansowany. Wartości głównie występują w przedziale [40;80]. Najczęściej występującą wartością dryblingu jest liczba 64.



Wnioski:

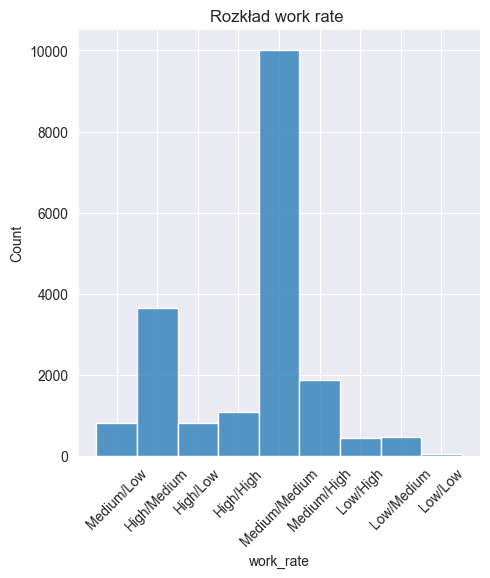
-Napastnicy posiadają bardzo wysoką korelacje między statystykami związanymi ze strzelaniem:

- Korelacja między statystyką shooting (strzelanie) oraz attacking\_finishing (wykończenie) wynosi 0.97.

- Shooting (strzelanie) silnie koreluje również z attacking\_volleys (woleje, czyli strzały z powietrza) – wartość 0.88.

-Korelacja attacking\_finishing (wykończenie) oraz attacking\_volleys (woleje) wynosi 0.90.

- Kolejną ciekawą korelację jest attacking\_finishing (wykończenie) od skill\_moves (ilość gwiazdek sztuczek) oraz attacking\_volleys (woleje) od skill\_moves (ilość gwiazdek sztuczek). Tak jak pokazałem na podstawie wcześniejszego wykresu, skill moves jest zależne od dryblingu. Natomiast dla napastników taka korelacje również występują ze szczególnymi umiejętnościami strzeleckimi.

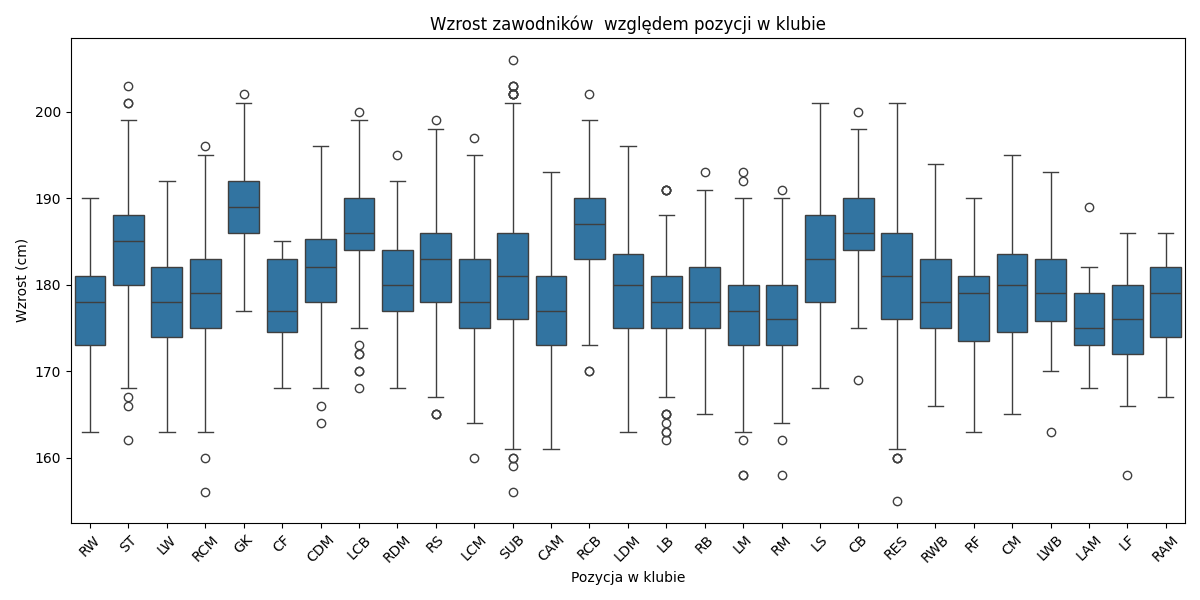


Wnioski

- z wykresu widać, że cechy nie są zbalansowane. Zdecydowana większość zawodników ma pracowitości Medium/Medium.

- Cecha Low/Low występuje bardzo rzadko, co jest dość logiczne bo mało piłkarzy jest słabo zaangażowanych zarówno w ataku jak i w obronie.

- Trzy najmniej liczne cechy kategorialne zawierają w obronie lub w ataku poziom Low, a trzy najbardziej liczne cechy kategorialne zawierają w obronie lub w ataku poziom Medium. Czyli dużo bardziej prawdopodobne jest wylosowanie piłkarza który będzie miał którąś z pracowitości na poziomie Medium niż Low.



Wnioski:

- Najwyżsi zawodnicy grają na pozycjach: bramkarz (GK) oraz środkowy obrońca (LCB, RCB oraz CB)

- Boczni pomocnicy oraz skrzydłowi mają najniższe statystyki wzrostu (LM, RM, LW, RW).

- Bardzo szerokim zakresem wzrostu charakteryzuje się środkowy napastnik (ST). Co może świadczyć o różnych klasach podziałów napastników ze względu na jego typ.

- W każdej grupie występują outliery, zarówno te wysokie jak i niskie.

- Wzrost nie zależy od strony na boisku. Jeśli porównamy wykresy LM z RM, LS z RS, LB z RB dochodzimy do wniosku, że wyglądają dość podobnie.

# MODELE MASZYNOWE BEZ OPTYMALIZACJI

Na przedstawionym zbiorze danym zostały wytrenowane modele. Obecnie nie zostały na nim wykorzystane żadne metody optymalizacji.

Wyniki pomiarów z zaokrągleniem do 6 cyfry po przecinku dla zbioru testowego

|  |  |  |
| --- | --- | --- |
| Nazwa modelu | R2 | MSE |
| Regresja Liniowa | 0.966358 | 1.58329 |
| Random Forest Regressor | 0.994103 | 0.27750 |
| SVR | 0.976053 | 1.12701 |

Wyniki pomiarów z zaokrągleniem do 13 cyfry po przecinku dla zbioru treningowego

|  |  |  |
| --- | --- | --- |
| Nazwa modelu | R2 | MSE |
| Regresja Liniowa | 0.9999998154854 | 0.00000874651 |
| Random Forest Regressor | 0.9992620307190 | 0.03498185303 |
| SVR | 0.9839096974959 | 0.76272632473 |

Częścią projektu jest również własna implementacji regresji liniowej.

Wyniki pomiarów z zaokrągleniem do 5 cyfr po przecinku

|  |  |  |
| --- | --- | --- |
| Nazwa modelu | R2 | MSE |
| Closed Form | 2.6555 | 0.9433 |
| Gradient Descent | 2.6503 | 0.9432 |
| sklearn | 2.6553 | 0.9433 |

Jak widać wyniki są bardzo zbliżone, co może wskazywać na poprawność mojej implentacji.

# OPTYMALIZACJA MODELI MASZYNOWYCH

**3-krotna walidacja krzyżowa**

W projekcie zastosowałem K-Fold Cross Validation, aby uzyskać bardziej wiarygodną ocenę skuteczności modelu. Dzięki tej technice każdy fragment danych mógł być zarówno częścią treningową, jak i testową, co pozwala lepiej wykorzystać dostępny zbiór.

Random Forest Regressor

|  |  |  |
| --- | --- | --- |
| Fold | R2 | RMSE |
| Fold 1 | 0.994 | 0.54 |
| Fold 2 | 0.994 | 0.52 |
| Fold 3 | 0.994 | 0.53 |

Linear Regression

|  |  |  |
| --- | --- | --- |
| Fold | R2 | RMSE |
| Fold 1 | 0.966 | 1.25 |
| Fold 2 | 0.966 | 1.27 |
| Fold 3 | 0.963 | 1.33 |

Support Vector Regression

|  |  |  |
| --- | --- | --- |
| Fold | R2 | RMSE |
| Fold 1 | 0.975 | 1.08 |
| Fold 2 | 0.973 | 1.12 |
| Fold 3 | 0.969 | 1.22 |

Linear Regression (własna implementacja)

|  |  |  |
| --- | --- | --- |
| Fold | R2 | MSE |
| Fold 1 | 0.9226 | 3.5907 |
| Fold 2 | 0.9227 | 3.4513 |
| Fold 3 | 0.96329 | 3.2143 |

Wnioski:

1. Wyniki modelu Random Forest Regressor są bardzo stabilne.
2. Wyniki modelu Linear Regression (oraz własnej jej implementacji), Support Vector Regression nie są tak stabilne jak RFR, jednak wykazują się całkiem dobrą stablinością.
3. Fold 3 często pokazuje gorsze wyniki w modelach gotowych (sklearn), co może wskazywać, że w tej części danych znajduje się więcej przypadków odstających lub trudniejszych do przewidzenia.

**Wykresy zbieżności**

Przypomnijmy wyniki dla regresji liniowej:

1. Na zbiorze treningowym:

-R² (trening): 0.9999998 - praktycznie idealne dopasowanie.

-MSE (trening): 0.00000875 - bardzo mały błąd.

b) Na zbiorze testowym:

-R² (test): 0.966358 - nadal bardzo dobre dopasowanie.

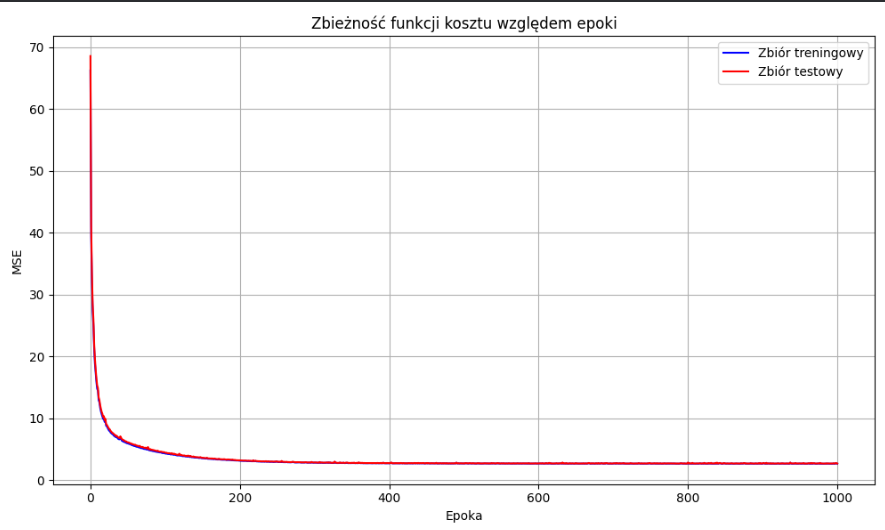
-MSE (test): 1.58329 - większy błąd.

Mamy tutaj pewien poziom overfittingu. Możemy polepszyć skuteczność naszego modelu poprzez zwiększenie złożoności modelu poprzez dodanie dodatkowych cech (np. PolynomialFeatures). Niestety przez ograniczenia sprzętowe nie jestem w stanie odpalić kodu dla stopnia 2 (screen z błędem poniżej), dlatego ograniczam model do 50 najważniejszych cech ('feature\_selection', SelectKBest(f\_regression, k=50)).

Wyniki prezentują się następująco:

|  |  |  |
| --- | --- | --- |
| Model | R2 | MSE |
| Regresja Liniowa (treningowy) | 0.9999998154854 | 0.00000874651 |
| Regresja Liniowa (testowy) | 0.966358 | 1.58329 |
| Regresja Liniowa st.2 (treningowy) | 0.9836033878126592 | 0.7772462791509372 |
| Regresja Liniowa st.2 (testowy) | 0.9774865369069581 | 1.0595563552205676 |

Jak widać ograniczenie modelu do 50 cech oraz zwiększenie złożoności do 2 stopnia poprawiły wyniki.

Teraz przeanalizuje czy w mojej własnej implementacji regresji liniowej występuje problem underfittingu lub overfittingu. Będzie to możliwe po analizie wykresu:  
Funkcja przedstawiona na wykresie jest malejąca oraz zbieżna.Na podstawie wykresu, można zauważyć, że krzywe dla zbioru treningowego (niebieska) i testowego (czerwona) są bardzo blisko siebie. Obie krzywe wykazują spadek wartości MSE na początku, a następnie stabilizują się na podobnym poziomie.

To sugeruje, że Overfitting nie występuje, ponieważ wtedy różnica między błędem na zbiorze treningowym a testowym byłaby duża (np. bardzo niski błąd na treningu i znacznie wyższy na teście). Underfitting również nie występuje, ponieważ funkcja kosztu spada i stabilizuje się na niskim poziomie, co oznacza, że model uczy się dobrze zarówno na danych treningowych, jak i testowych.

**Dodanie regularyzacji L1 i L2**

W celu dalszych usprawnień modelu dodam regularyzację. Najpierw przeanalizujmy wpływ dodania metody Lasso (L1) na wyniki modelu, a następnie wpływ metody Ridge (L2).

Wyniki bez regularyzacji:

|  |  |  |
| --- | --- | --- |
| Model | R2 | MSE |
| Regresja Liniowa st.2 (treningowy) | 0.9836033878126592 | 0.7772462791509372 |
| Regresja Liniowa st.2 (testowy) | 0.9774865369069581 | 1.0595563552205676 |

Wyniki z regularyzacją L1:

|  |  |  |
| --- | --- | --- |
| Model | R2 | MSE |
| Regresja Liniowa st.2 (treningowy) | 0.9781942004151016 | 1.033657221237361 |
| Regresja Liniowa st.2 (testowy) | 0.9764385733701784 | 1.1088769071430105 |

Lasso działało dla parametru alpha=0.01. Wyniki nie poprawiają się. Po próbie zmianie parametru alpha wyniki nie poprawiały się

Wyniki z regularyzacją L2:

|  |  |  |
| --- | --- | --- |
| Model | R2 | MSE |
| Regresja Liniowa st.2 (treningowy) | 0.9834776581371123 | 0.7832062251069201 |
| Regresja Liniowa st.2 (testowy) | 0.9778374864099298 | 1.0430395370527692 |

Ridge działał również dla parametru alpha=0.01. Testowałem również wartość 10 oraz 100 razy większą jednak obie dały gorszy wynik. Używając Ridge udało się delikatnie poprawić wyniki.

Lasso (L1) i Ridge (L2) różnią się sposobem regularyzacji: Lasso dodaje do funkcji kosztu sumę wartości bezwzględnych współczynników, a Ridge – sumę ich kwadratów. Lasso może zerować współczynniki, co prowadzi do automatycznej selekcji cech, natomiast Ridge jedynie zmniejsza ich wartości, ale ich nie usuwa.

W tabeli prezentują się pierwsze 10 wag, co pozwala porównać jak działają te metody

|  |  |  |  |
| --- | --- | --- | --- |
|  | Bez regularyzacji | Lasso | Ridge |
| W0 | 0 | 0 | 0 |
| W1 | -0.0322 | -0.0653 | -0.0450 |
| W2 | 1.5211 | 2.5176 | 1.5794 |
| W3 | 3.0202 | 1.7012 | 2.9031 |
| W4 | 0.1295 | 0.0943 | 0.0708 |
| W5 | 1.4937 | 2.1721 | 1.5615 |
| W6 | -0.0133 | 0 | -0.0263 |
| W7 | -0.0341 | 0 | -0.0367 |
| W8 | -0.9420 | 0 | -0.9214 |
| W9 | -0.2881 | 0 | -0.3201 |

Dodałem również regularyzację dla mojej własnej implementacji regresji liniowej. Porównując wyniki nie widać żadnej poprawy, wykresy bez i z regularyzacją wyglądają niemal identyczne. Użyłem poniższego wzoru do policzenia regularyzacji:



Źródło: <https://neptune.ai/blog/fighting-overfitting-with-l1-or-l2-regularization>

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Wykres, z którego metoda wykorzystuje regularyzacje

**Balansowanie zbiorów**

Wyniki modeli Random Forest Classifier, Logistic Regression oraz SVC bez, z użyciem oversampling oraz undersampling.

Random Forest Classifier

|  |  |  |  |
| --- | --- | --- | --- |
| Balansowanie | precision | recall | F1-score |
| brak | 0.4628 | 0.5489 | 0.4518 |
| Oversampling | 0.4969 | 0.5364 | 0.5039 |
| Undersampling | 0.5822 | 0.2599 | 0.3086 |

Logistic Regression

|  |  |  |  |
| --- | --- | --- | --- |
| Balansowanie | Precision | recall | F1-score |
| brak | 0.4969 | 0.5364 | 0.5089 |
| Oversampling | 0.5054 | 0.4896 | 0.4966 |
| Undersampling | 0.5405 | 0.2890 | 0.3424 |

SVC

|  |  |  |  |
| --- | --- | --- | --- |
| Balansowanie | precision | recall | F1-score |
| brak | 0.4793 | 0.5587 | 0.4729 |
| Oversampling |  |  |  |
| Undersampling | 0.5943 | 0.2718 | 0.3207 |