Kamil Tatrocki 280506

projekt z metod systemowych i decyzyjnych

SPIS TREŚCI

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.

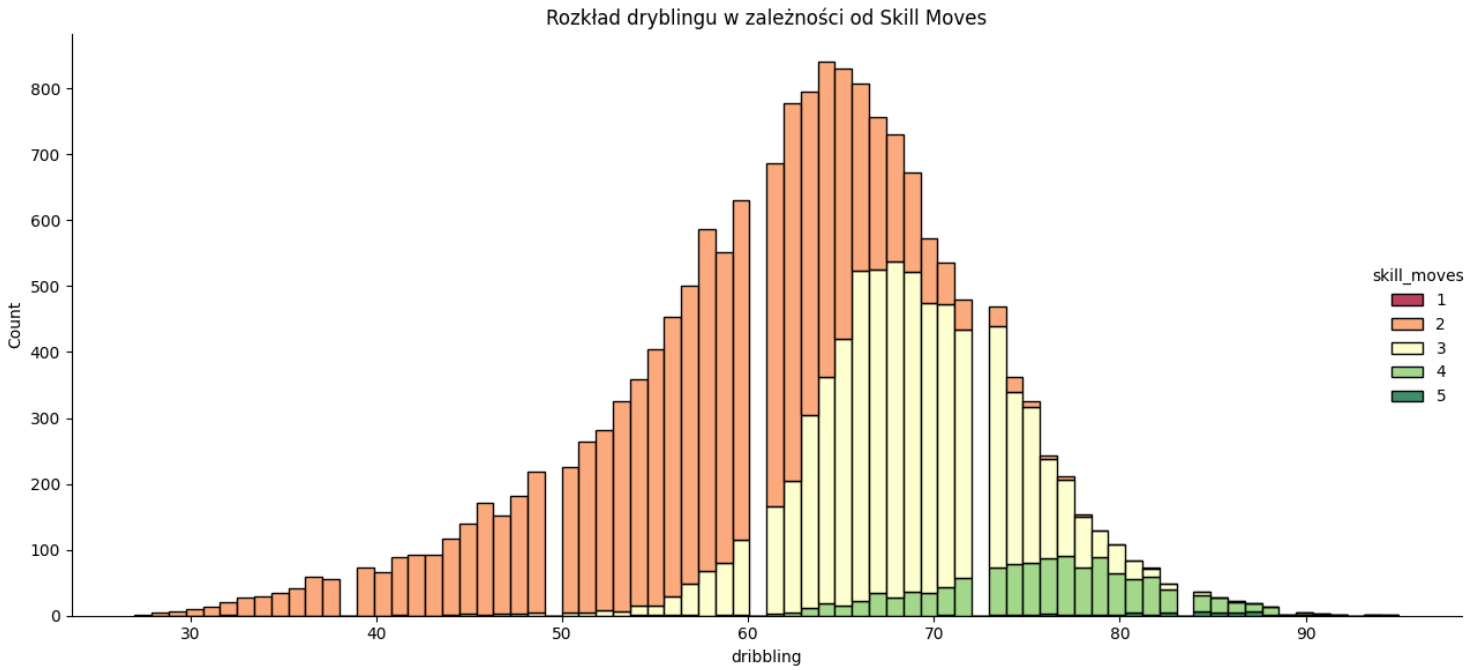
EKSPLORACJA ZBIORU DANYCH

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.

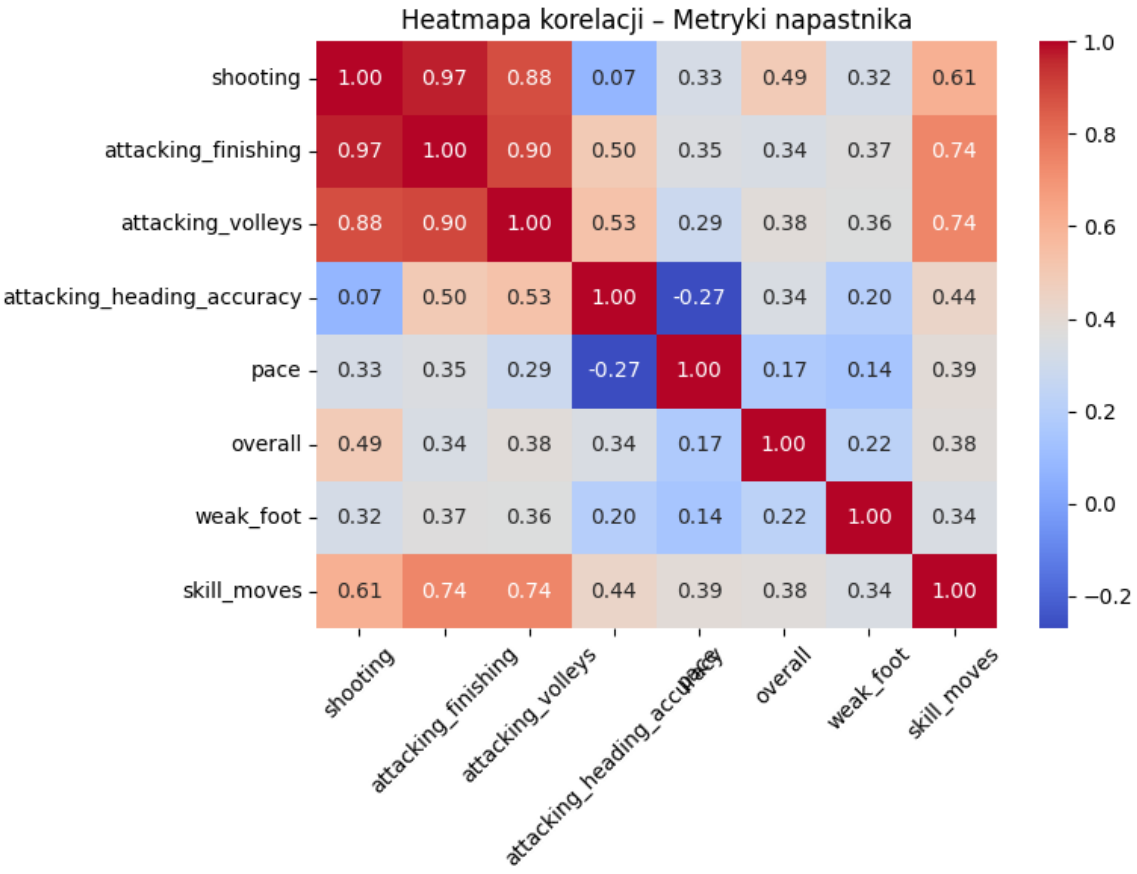


Wnioski:

- 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:

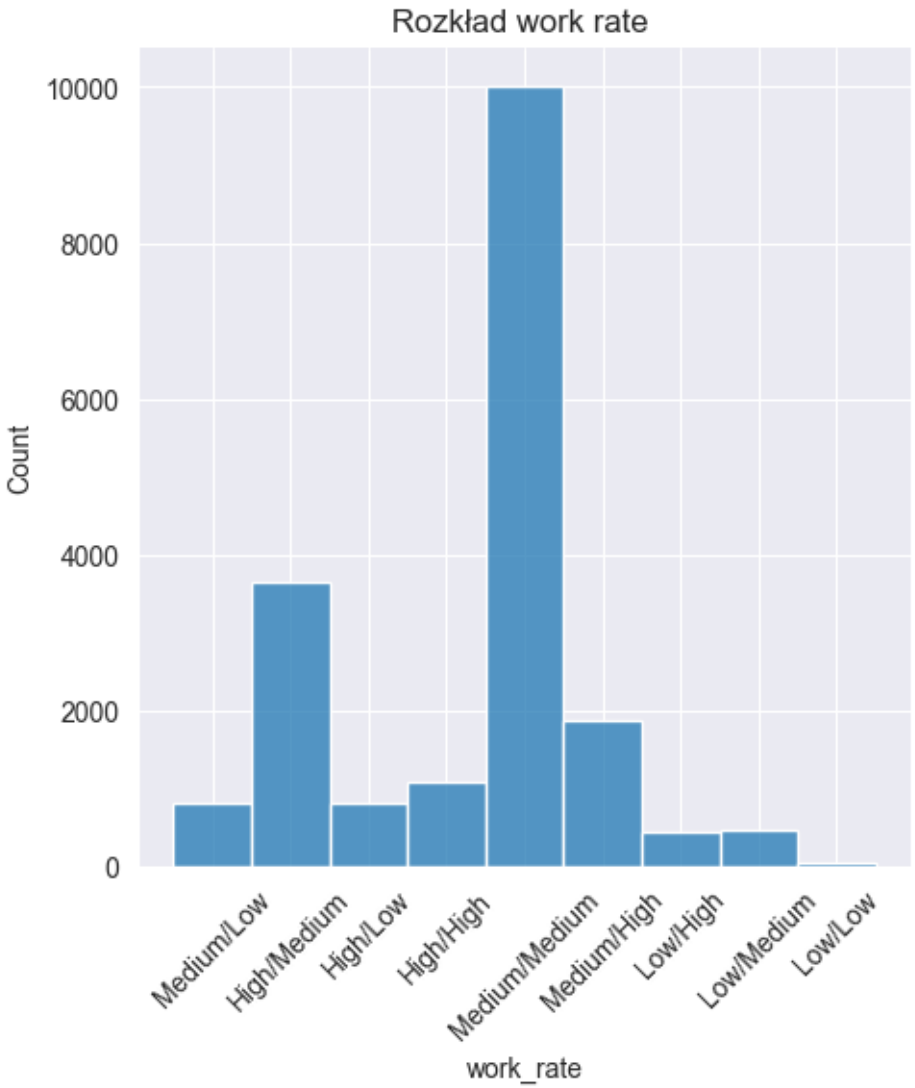
-Napastnicy posiadają bardzo wysoką korelacje między statystykami związanymi ze strzelaniem:

- 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.

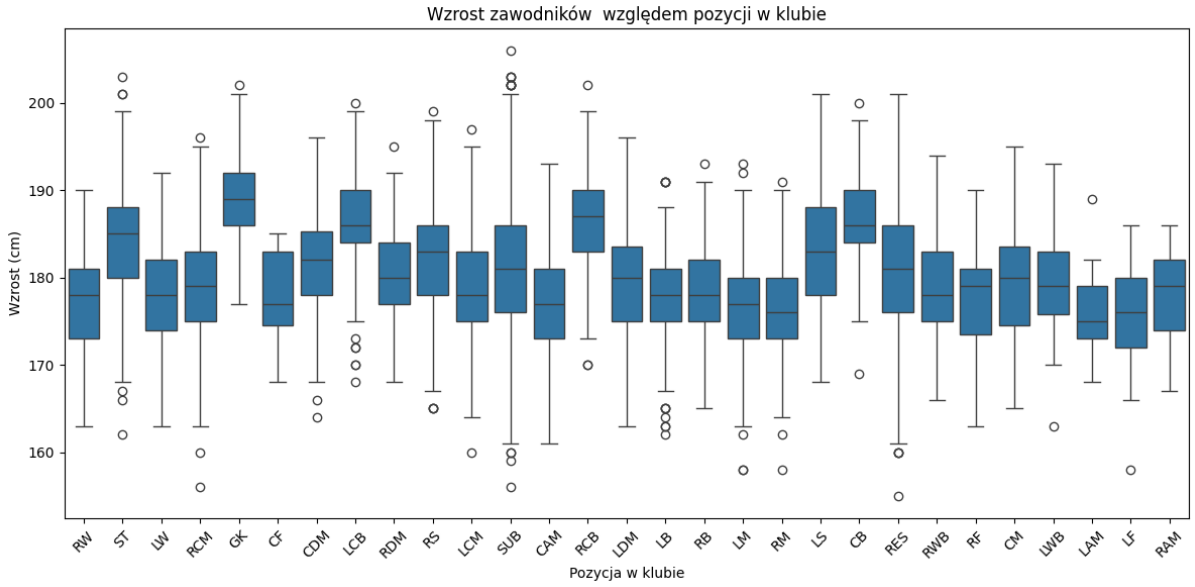


Wnioski

- 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 implementacja 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 implementacji.

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 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. (własna implementacja regresji liniowej pracowała na innych foldach niż pozostałe modele, także do tego modelu ta uwaga się nie dotyczy)

Wykresy zbieżności

Przypomnijmy wyniki dla regresji liniowej:

a) Na zbiorze treningowym:
-R^2 (trening): 0.9999998 - praktycznie idealne dopasowanie.
-MSE (trening): 0.00000875 - bardzo mały błąd.

b) Na zbiorze testowym:

-R^2 (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)).

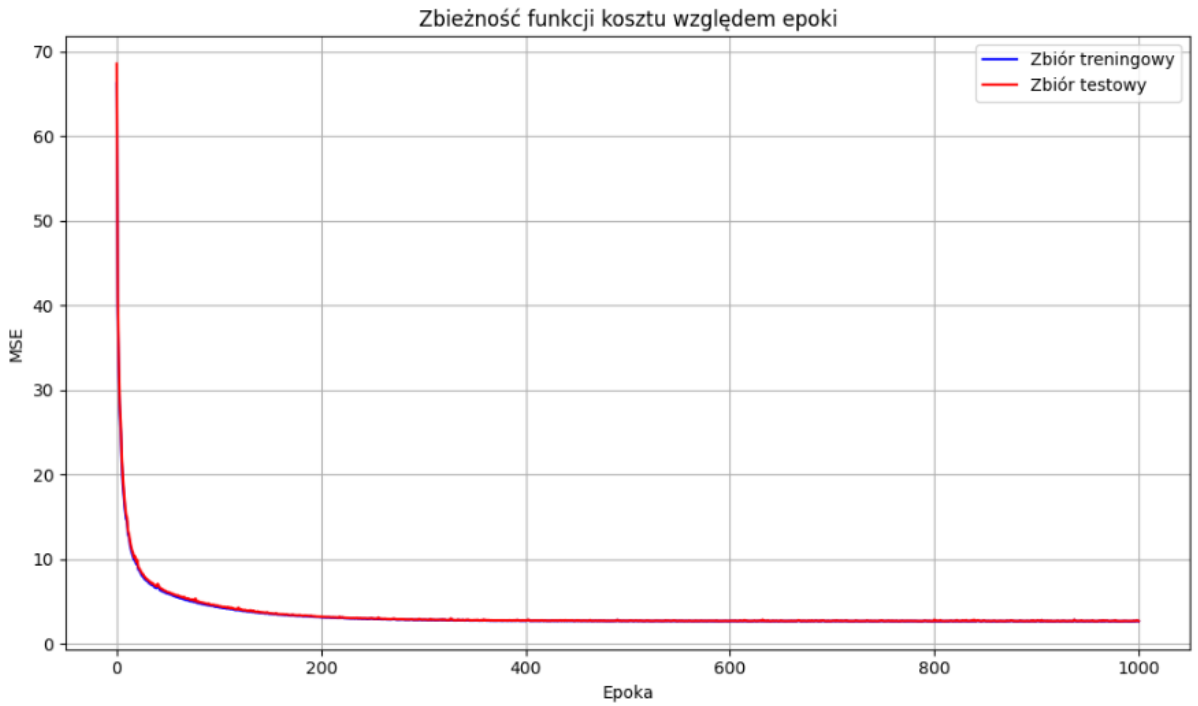
```
numpy._core._exceptions._ArrayMemoryError: Unable to allocate 7.89 GiB for an array with shape (1058713120,) and data type uint64
```

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 dodałem 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

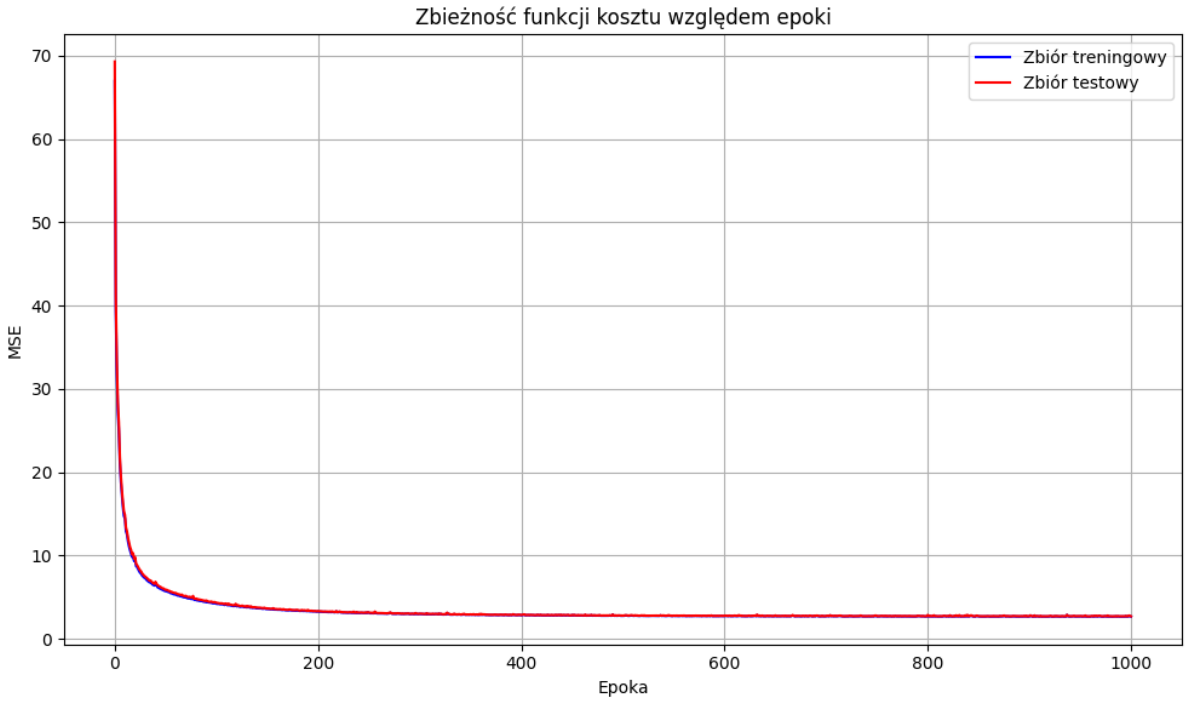
	Bez regularyzacji	Lasso	Ridge
W0	0	0	0
W1	-0.0322	-0.0653	-0.0450
W2	1.5211	2.5176	1.5794
W3	3.0202	1.7012	2.9031
W4	0.1295	0.0943	0.0708
W5	1.4937	2.1721	1.5615
W6	-0.0133	0	-0.0263
W7	-0.0341	0	-0.0367
W8	-0.9420	0	-0.9214
W9	-0.2881	0	-0.3201

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:

$$LossFunction = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y} - Y)^2 + \lambda \sum_{i=1}^{N} \theta_i^2$$

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

Wykres, z którego metoda wykorzystuje regularyzację



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	0.5175	0.5294	0.5195
Undersampling	0.5943	0.2718	0.3207

Wnioski:

- Dla Random Forest najlepsze rezultaty uzyskano przy użyciu oversamplingu. Undersampling wyraźnie obniżył skuteczność modelu, głównie przez drastyczny spadek recall.

- Logistic Regression najlepiej radzi sobie bez żadnego balansowania danych. Oversampling daje podobne wyniki, natomiast undersampling zdecydowanie pogarsza jakość klasyfikacji.

- Podobnie jak w Random Forest, oversampling poprawia wyniki SVC. Undersampling negatywnie wpływa na skuteczność modelu.

Optymalizacja hiperparemetrów.

Przeszukiwanie hiperparametrów jest złożonym zadaniem, ponieważ liczba możliwych kombinacji rośnie wykładniczo wraz ze wzrostem liczby hiperparametrów, co może prowadzić do bardzo długiego czasu obliczeń i trudności w znalezieniu optymalnej kombinacji.

W mojej implementacji wybrałem dwa modele regresji: Random Forest Regressor oraz Support Vector Regressor, stosując technikę GridSearchCV. Z powodu ograniczeń sprzętowych ograniczyłem się jedynie do dwóch cech hiperparametrów w każdym modelu, po trzy różne wartości dla każdej cechy, co i tak skutkowało trwającymi około 3 godzin obliczeniami. Dla Random Forest Regressora analizowałem parametry n_estimators oraz max_depth. Parametr n_estimators określa liczbę drzew decyzyjnych używanych przez model, wpływając na stabilność i precyzję predykcji (więcej drzew zazwyczaj poprawia wyniki, ale kosztem czasu obliczeń). Z kolei parametr max_depth kontroluje maksymalną głębokość każdego drzewa, co wpływa na zdolność modelu do generalizacji (płytsze drzewa mogą zapobiegać przeuczeniu, ale mogą też niedostatecznie dopasować się do danych). W przypadku Support Vector Regressora badałem wpływ parametrów kernel, C oraz epsilon. Kernel określa sposób transformacji przestrzeni cech (w tym przypadku użyłem tylko kernela typu RBF, czyli radialnego), parametr C kontroluje balans pomiędzy minimalizacją błędu treningowego a maksymalizacją marginesu decyzyjnego (wyższe wartości C powodują mocniejsze dopasowanie do danych treningowych), a epsilon określa zakres tolerowanego błędu predykcji, wpływając na dopuszczalną szerokość marginesu regresji.

Wyniki optymalizacji hiperparametrów przedstawiają się następująco:

A) **Random Forest Regressor**

Analizowane parametry:

'reg__n_estimators': [100, 200, 400],
'reg__max_depth': [None, 10, 20]

Najlepsze okazały się parametry: {'reg__max_depth': 20, 'reg__n_estimators': 400} i dały następujące wyniki:

	R2	MSE
Zbiór treningowy	0.999	0.0
Zbiór testowy	0.994	0.3

Wnioski:

Random Forest Regressor bardzo dobrze poradził sobie z problemem regresyjnym, oferując zarówno wysoką dokładność, jak i stabilność predykcji. Ryzyko przeuczenia jest niewielkie, co potwierdzają porównywalne wyniki na zbiorze treningowym i testowym.

B) **Support Vector Regression**

Analizowane parametry:

'reg__C': [1, 10, 100],
'reg__epsilon': [0.1, 0.2, 0.5]

Najlepsze okazały się parametry: {'reg__C': 100, 'reg__epsilon': 0.1, 'reg__kernel': 'rbf'} i dały następujące wyniki:

	R2	MSE
Zbiór treningowy	1.000	0.0
Zbiór testowy	0.987	0.6

Wnioski:

SVR również okazał się skutecznym modelem, jednak nieco gorsze wyniki na zbiorze testowym mogą sugerować, że model bardziej dopasował się do danych treningowych, kosztem uogólnienia. Może być bardziej wrażliwy na nadmierne dopasowanie.

Metody Ensemble

Model	MSE
VotingRegressor (testowy)	0,98924
VotingRegressor (treningowy)	0,997936
StackingRegressor (testowy)	0,994521
StackingRegressor (treningowy)	0,999314

Wnioski:

- Dla Voting Regressor MSE na zbiorze testowym wynosi 0,98924, natomiast na zbiorze treningowym wynosi 0,997936. Wartości są bardzo zbliżone, co sugeruje, że model jest stabilny i dobrze uogólnia dane.

- Dla Stacking Regressor MSE na zbiorze testowym wynosi 0,994521, a na treningowym 0,999314. Różnice również są niewielkie, choć nieco wyższe niż w przypadku VotingRegressora, co sugeruje minimalnie słabsze uogólnienie danych.

- Brak znaczącej różnicy między wynikami MSE na zbiorach testowych i treningowych dla obu modeli sugeruje brak wyraźnego przeuczenia (overfitting) czy niedouczenia (underfitting).

STUDIUM ABLACYJNE DLA REGRESJI LINIOWEJ

Niniejsza część ma na celu podsumowanie wszystkich prób optymalizacji i pokazanie tylko tych metod, które dały najlepsze wyniki. Poniższe tytuły tabel są niejako krokami, które były dodawane krok po kroku w celu optymalizacji analizowanego modelu.

Wyniki bez żadnych optymalizacji

Nazwa modelu	R2	MSE
Regresja Liniowa (zbiór treningowy)	0.9999998154854	0.00000874651
Regresja Liniowa (zbiór testowy)	0.966358	1.58329

Walidacja krzyżowa

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

Ograniczenie do 50 cech oraz dodanie PolynomialFeatures

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

Dodanie regularyzacji R2

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

Na podstawie przedstawionego studium ablacyjnego dla regresji liniowej można zauważyć, że początkowy model bez żadnych optymalizacji osiągał niemal idealne dopasowanie do zbioru treningowego ($R^2 \approx 1$), jednak jego skuteczność na zbiorze testowym była już znacznie niższa ($R^2 \approx 0.966$), co sugeruje zjawisko przeuczenia. Walidacja krzyżowa wykazała stabilne, choć nieco zróżnicowane wyniki (R^2 w granicach 0.963–0.966), co potwierdza, że model dobrze radzi sobie z danymi, ale może mieć trudności z uogólnianiem. Wprowadzenie ograniczenia liczby cech do 50 oraz dodanie cech wielomianowych przyniosło znaczną poprawę wyników na zbiorze testowym – współczynnik determinacji wzrósł do około 0.977, a błąd MSE się zmniejszył. Ograniczenie liczby cech zredukowało ryzyko nadmiernego dopasowania, a cechy wielomianowe pozwoliły uchwycić bardziej złożone, nieliniowe zależności między zmiennymi. Dodanie regularyzacji nie wpłynęło drastycznie na wartość R^2, ale pozwoliło jeszcze bardziej ustabilizować model poprzez niewielkie obniżenie błędu MSE na zbiorze treningowym. Dzięki regularyzacji model lepiej radzi sobie z potencjalnym szumem w danych i zmniejsza nadmierne dopasowanie do danych treningowych. Różnica pomiędzy wynikami na zbiorze treningowym i testowym po regularyzacji stała się bardzo mała, co świadczy o dobrej zdolności modelu do generalizacji. Choć każdy krok optymalizacji nie zawsze przynosił wyraźną poprawę wskaźników jakości, to kolejne modyfikacje prowadziły do stworzenia bardziej odpornego i przewidywalnego modelu. Finalna wersja modelu, zawierająca ograniczenie liczby cech, cechy wielomianowe oraz regularyzację, zapewnia najlepszy kompromis między dokładnością a zdolnością do przewidywania na nieznanych danych.

NAJLEPSZY MODEL

Random Forest Regressor reg__max_depth = 20, reg__n_estimators = 400

	R2	MSE
Zbiór treningowy	0.999	0.0
Zbiór testowy	0.994	0.3

Najlepszym modelem do przewidywania kategorii „overall” okazał się Random Forest Regressor z parametrami max_depth=20 oraz n_estimators=400. Na podstawie uzyskanych wyników można stwierdzić, że model ten charakteryzuje się bardzo wysoką dokładnością predykcji zarówno na zbiorze treningowym ($R^2 = 0,999$; MSE = 0,0), jak i na zbiorze testowym ($R^2 = 0,994$; MSE = 0,3). Tak wysoka skuteczność modelu może wynikać z tego, że Random Forest potrafi dobrze uchwycić zarówno liniowe, jak i nieliniowe zależności pomiędzy dużą liczbą zmiennych a zmienną docelową. Dzięki podziałowi na wiele drzew decyzyjnych, Random Forest jest mniej podatny na przeuczenie, co potwierdza bardzo wysoka, choć minimalnie niższa skuteczność na zbiorze testowym.

Sukces tego modelu jest także wynikiem trafnego wyboru parametrów – ograniczenie głębokości drzew (max_depth=20) zapobiega ich nadmiernemu przeuczeniu i zapewnia generalizację, natomiast wysoka liczba drzew (n_estimators=400) gwarantuje stabilność wyników oraz skuteczne radzenie sobie ze złożonymi relacjami między danymi.