AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Informatyki, Elektroniki i Telekomunikacji

Projekt dyplomowy

Zastosowanie metod uczenia maszynowego do przewidywania wyników plebiscytu sportowego na najbardziej wartościowego zawodnika

Application of Machine Learning Methods for Most Valuable Player Forecasting

Autor:	*Piotr Lehmann*
Kierunek studiów:	*Teleinformatyka*
Opiekun pracy:	*dr inż. Michał Grega*

Kraków, 2024

Spis treści

Wstęp

W dobie znacznego wzrostu popularności sztucznej inteligencji, zupełnie naturalną stała się chęć wykorzystania jej potencjału w coraz to nowszych dziedzinach. Najciekawsze wydają się te z nich, które przez lata wygenerowały i nadal generują ogrom różnorodnych danych, pozwalając na ich dogłębną analizę, odkrywanie ukrytych wzorców, czy przewidywanie określonych fenomenów i wydarzeń. Jedną z najpopularniejszych tego typu dziedzin jest sport, a już zwłaszcza ten na najwyższym poziomie.

Dzięki dynamicznemu rozwojowi i bogactwu dostępnych danych stanowi on idealne pole do zastosowania i rozwoju algorytmów uczenia maszynowego. Wyniki meczów, rezultaty turniejów, liczba fauli czy zdobytych punktów – to typowe przykłady danych, które mogą być analizowane i przewidywane właśnie za ich pomocą. Nieco mniej oczywiste, choć równie interesujące, wydaje się być przewidywanie wyników plebiscytów sportowych. Budzą one przecież emocje i kontrowersje równe wydarzeniom na boisku. Niniejsza praca koncentruje się właśnie na zastosowaniu metod uczenia maszynowego do przewidywania wyników takiego plebiscytu.

Wybraną dyscypliną jest koszykówka, a konkretniej skupiono się na lidze *NBA*, czyli najpopularniejszej lidze koszykarskiej na świecie. Władze ligi co roku, pod koniec sezonu, wręczają najwybitniejszym zawodnikom nagrody za indywidualne osiągnięcia. Najważniejszą z tych nagród, jest statuetka dla najbardziej wartościowego zawodnika sezonu. Jest on wybierany na drodze głosowania przez, niezależnych ekspertów i po uhonorowaniu uważany za najbardziej wszechstronnego, dominującego oraz generalnie najlepszego zawodnika sezonu.

Celem projektu opisanego w pracy, było opracowanie narzędzia opartego na algorytmach uczenia maszynowego, które otrzymując na wejściu statystyki wszystkich zawodników z wybranego sezonu, będzie w stanie utworzyć prognozowany ranking najlepszych 5 koszykarzy, wybierając przy tym przede wszystkim, przewidywanego zwycięzcę plebiscytu na najbardziej wartościowego zawodnika. Aby osiągnąć założony cel, implementowano, testowano, wyciągano wnioski i na ich podstawie przystosowywano narzędzie.

Praca szczegółowo opisuje chronologicznie uszeregowane działania autora, podkreślając kolejne etapy realizacji projektu i wizualizując osiągnięte efekty. Opisano między innymi proces tworzenia zbioru danych oraz trenowania na nim modelu regresyjnego. W celu zwiększenia skuteczności narzędzia, zaimplementowano własne adaptacje, takie jak filtry progowe i własny hiperparametr. Zaimplementowano również specyficzną, dopasowaną do problemu funkcję straty, niedostępną w podstawowych bibliotekach. Ostatecznie w pracy zawarto wyniki gotowego narzędzia, oceniając je w wymierny sposób i porównując z prawdziwymi wynikami plebiscytów na najbardziej wartościowego zawodnika *NBA*. Docelowym zadaniem narzędzia było jednak przewidywanie wyników głosowania jeszcze przed ich oficjalnym ogłoszeniem. Z tego

względu, praca kończy się rozdziałem poświęconym ocenie jego skuteczności na podstawie predykcji najbardziej wartościowego zawodnika sezonu 2024-25, który w chwili dokonywania prognoz jeszcze się nie zakończył.

1. Wprowadzenie teoretyczne

W tym rozdziale omówione zostały kluczowe pojęcia oraz metody związane z problemem prognozowania najbardziej wartościowego gracza NBA. Przedstawiono charakterystykę plebiscytu MVP oraz wyjaśniono istotne aspekty i zagadnienia związane z procesem wyboru laureata. Opisano również wybrane metody uczenia maszynowego, ze szczególnym uwzględnieniem regresji lasu losowego, oraz omówiono podstawowe techniki oceny modeli predykcyjnych. Na końcu dokonano przeglądu literatury związanej z dotychczasowymi podejściami do prognozowania wyników plebiscytu na najbardziej wartościowego gracza.

1.1. Plebiscyt MVP w NBA

NBA (*National Basketball Association*) to najpopularniejsza i największa liga koszykówki na świecie, założona w 1946 roku. Uznawana jest za najbardziej prestiżową organizację koszykarską, skupiającą najlepszych zawodników na świecie. Liga składa się z 30 drużyn, a w każdym sezonie w rozgrywkach bierze udział około 500 zawodników. Ze względu na liczne wyróżnienia przyznawane co roku, tworzona i monitorowana jest obszerna baza wcześniej zebranych statystyk indywidualnych i drużynowych. Jest ona udostępniana i aktualizowana codziennie na oficjalnej stronie ligi NBA [1].

Sezon regularny - jest główną częścią pełnego cyklu rozgrywek NBA, w którym każda z drużyn od października do kwietnia, rozgrywa około 80 meczy. Jest to kluczowy etap, podczas którego zespoły rywalizują o jak najwyższe miejsce w tabeli, co zapewnia im korzystniejsze rozstawienie w drabince pucharowej (fazie play-off). Na podstawie wyników i osiągnięć w sezonie regularnym wyłaniani są także laureaci nagród indywidualnych, z których najbardziej prestiżową jest statuetka MVP.

MVP (*Most Valuable Player*) - w sporcie, najbardziej wartościowym graczem z reguły nazywamy sportowca znacznie wyróżniającego się na tle aktywnych zawodników w danym okresie - w tym przypadku w pojedynczym sezonie regularnym NBA. W trakcie sezonu każdy gracz generuje szereg pozytywnych i negatywnych, osobistych statystyk w trakcie rozgrywania meczów, które są po każdym z nich zbierane, aktualizowane i publikowane na oficjalnej stronie NBA [1]. Statystyki te obejmują aspekty gry zarówno ofensywnej, jak i defensywnej, takie jak liczba zdobytych punktów, zbiórek, asyst, przechwytów czy bloków. W tym momencie istnieje 269 takich statystyk[1], ich liczba może jednak zmieniać się w kolejnych sezonach. Niektóre z nich to złożone wskaźniki powstające z połączenia innych danych – np. wskaźnik efektywności gry (*Player Efficiency Rating*). Analizując jakościowo dotychczasowych laureatów, można

[1] Strona NBA Glossary zbierająca i wyjaśniająca wszystkie statystyki zbierane przez NBA

stwierdzić, że historycznie gracze otrzymujący tytuł MVP wyróżniali się nie tylko indywidualnymi osiągnięciami, ale także istotnym wkładem w sukces swojej drużyny. Na ocenę końcową składają się zatem zarówno osobiste statystyki zawodnika, jak i jego wpływ na osiągnięcia zespołu. Wszystko oczywiście w porównaniu do pozostałych pretendentów [2]. Po zakończeniu sezonu regularnego, MVP wybierany jest drogą głosowania. Grupa niezależnych ekspertów, niezwiązanych z żadnym z zawodników ani drużyn, oddaje subiektywne głosy, wybierając ich zdaniem pięciu najlepszych graczy sezonu, opierając się na wcześniej wspomnianych kryteriach. Miejsca są nagradzane punktami w następujący sposób: 10 punktów za pierwsze, 7 punktów za drugie, 5 punktów za trzecie, 3 punkty za czwarte i 1 punkt za piąte miejsce. Sumując głosy dla każdego aktywnego gracza, tworzy się ranking MVP, a koszykarz z największą liczbą zebranych punktów zostaje ogłoszony najbardziej wartościowym graczem danego sezonu. Wygrana w plebiscycie dostarcza ogromnego prestiżu, porównywalnego z uzyskaniem Złotej Piłki w piłce nożnej, stanowiąc najwyższe wyróżnienie dla zawodnika.

Choć sama nagroda nie wiąże się bezpośrednio z korzyściami finansowymi, koszykarz, który zdobywa ten tytuł, staje się niezwykle pożądany na rynku transferowym, co zwiększa jego wartość w oczach drużyn poszukujących zmian w składzie. Dodatkowo, zawodnicy świadomi swojej szansy na zdobycie statuetki często negocjują w kontraktach klauzule, które przewidują dodatkowe wynagrodzenie w przypadku jej wygrania. Przykładem może być tutaj Nick Collison z drużyny Seattle Supersonics, którego nazwisko figurowało w gronie faworytów do tytułu MVP. Dzięki temu mógł wynegocjować klauzulę, która przyznawała mu 250 000 dolarów w razie wygranej plebiscytu MVP.

1.2. Wprowadzenie do uczenia maszynowego

Uczenie maszynowe (ang. *Machine Learning, ML*) jest jednym z najistotniejszych obszarów sztucznej inteligencji, który pozwala systemom komputerowym uczyć się i doskonalić na podstawie doświadczenia zbieranego podczas treningu, aby rozwiązać zadania określonego typu. W ramach tego obszaru, maszyny są w stanie wykrywać wzorce i podejmować decyzje na podstawie pokazanym im danych, co sprawia, że uczenie maszynowe staje się kluczową technologią, coraz częściej używaną w dzisiejszych czasach [3].

Rodzaje uczenia maszynowego

Uczenie maszynowe dzieli się na dwie główne kategorie, w zależności od charakterystyki zbioru danych i celów jakie ma osiągnąć model.

Uczenie nadzorowane

Uczenie nadzorowane (ang. *Supervised Learning*) jest jedną z najczęściej stosowanych metod uczenia maszynowego, polegającą na trenowaniu modelu oznakowanymi, inaczej mówiąc etykietowanymi danymi. Oznacza to, że dla każdego przykładu w zbiorze danych znana jest również odpowiedź (etykieta), którą model ma poprawnie przewidzieć. Celem jest nauczenie maszyny zależności pomiędzy danymi wejściowymi a odpowiadającymi im wyjściami, a przykładem zastosowania mogą być klasyfikacja (np. klasyfikowanie wiadomości jako spam lub nie-spam) oraz regresja (np. przewidywanie wartości liczbowych, takich jak ceny mieszkań).

Model uczony w sposób nadzorowany może starać się znaleźć sposób na precyzyjne przewidywanie wyników na podstawie danych na wiele sposobów w zależności od struktury i charakterystyki. W wielu przypadkach (np. bardzo popularnych sieciach neuronowych lub regresji liniowej) oznacza to znalezienie funkcji f, która odwzorowuje zależności między zmiennymi wejściowymi a wyjściowymi. Matematyczne można by to przedstawić za pomocą wzoru (1.1).

$$y = f(\mathbf{X}) = f([x_1, x_2, \ldots, x_n]) \tag{1.1}$$

Dane wejściowe są przedstawione w postaci $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, będąc wektorem cech/parametrów zawartych w danych. Wektorem cech mogą być między innymi: zestaw liczb reprezentujących dane numeryczne, tekst przekształcony na wektory liczb (np. za pomocą technik takich jak *Word2Vec*), obraz przekształcony do macierzy wartości odpowiadających jego pikselom, a także dane z innych dziedzin, takich jak dane dźwiękowe czy sygnały czasowe. Funkcja f powinna odwzorowywać dane wejściowe na wynik $\mathbf{y}$, który może przyjąć różną formę, w zależności od zadania.

Celem jest wytrenowanie funkcji f w taki sposób, aby jak najlepiej odwzorowywała zależności między danymi wejściowymi a wynikami wyjściowymi. Ocena jakości odwzorowania, obliczana jest na podstawie funkcji kosztu (*ang. Cost Function*), która pełni rolę narzędzia służącego do karania lub nagradzania modelu, w zależności od tego jak bliskie są jego przewidywania rzeczywistym wartościom. Dzięki wartości funkcji kosztu, model jest w stanie iteracyjnie dostosowywać swoje parametry, minimalizując różnicę między przewidywaniami a rzeczywistymi wynikami. Odbywa się to podczas procesu propagacji wstecznej (*ang. Backpropagation*), w którym gradient funkcji kosztu względem parametrów modelu jest obliczany i wykorzystywany do ich aktualizacji. Gradient wskazuje kierunek największego wzrostu funkcji kosztu, a dzięki metodom optymalizacji, takim jak *stochastic gradient descent*, parametry są aktualizowane w kierunku przeciwnym, co prowadzi do minimalizacji funkcji kosztu.

Warto jednak podkreślić, że nie wszystkie modele nadzorowanego uczenia maszynowego wymagają tworzenia jednej wyraźnej funkcji odwzorowującej ani stosowania procesu propagacji wstecznej, w ramach którego obliczany jest gradient funkcji kosztu. Inne podejście stosują np. maszyny wektorów nośnych (*ang. Support Vector Machines*), które minimalizują tą funkcję poprzez znalezienie hiperpłaszczyzny maksymalizującej margines między klasami. Dla odmiany, w modelach takich jak np. Lasy Losowe (*ang. Random Forests*), minimalizacja wartości funkcji kosztu opiera się o tworzenie wielu drzew decyzyjnych, działających niezależnie od siebie. Dochodzi następnie do agregacji ich wyników, a ich uśrednienie pozwala Lasom Losowym na minimalizację błędów oraz zwiększenie precyzji. Więcej informacji na temat *Random Forests* znajduje się w sekcji 1.3.

Uczenie nienadzorowane

Uczenie nienadzorowane (ang. *Unsupervised Learning*) różni się od uczenia nadzorowanego tym, że w tym przypadku dane nie zawierają etykiet. Celem jest znalezienie struktury lub wzorców w danych, np. poprzez ich klasteryzację. Uczenie nienadzorowane jest szczególnie przydatne, gdy etykietowanie danych jest kosztowne lub niemożliwe. Przykładem takich algorytmów jest analiza skupień (np. *k-means*). Ze względu na brak etykiet, funkcje straty w uczeniu nienadzorowanym często opierają się na miarach podobieństwa lub różnicy między danymi. Przykłady funkcji straty to miary odległości, takie jak odległość euklidesowa, czy miara entropii, które pomagają modelowi w grupowaniu podobnych danych lub np. wykrywaniu anomalii. Celem jest zatem również znalezienie wzorca, ale w tym przypadku model nie jest ukierunkowany na przewidywanie konkretnych wyników.

Zastosowania uczenia maszynowego

Uczenie maszynowe znajduje szerokie zastosowanie w różnych dziedzinach, od diagnostyki medycznej po systemy rekomendacji. Wśród najczęstszych obszarów wykorzystania można wymienić:

- **Cyberbezpieczeństwo** - wykrywanie zagrożeń, anomalii i złośliwego oprogramowania,
- **Internet rzeczy (IoT)** - przewidywanie zużycia energii, optymalizacja ruchu miejskiego,
- **E-commerce** - systemy rekomendacji i personalizacja ofert, reklam,
- **Opieka zdrowotna** - diagnozowanie chorób i przewidywanie ryzyka zachorowania,
- **Rozpoznawanie obrazów** - klasyfikacja i detekcja obiektów,
- **Regresja** - przewidywanie wartości ciągłych, takich jak ceny nieruchomości, prognozy temperatury lub wyników sportowych, czy analizy finansowe

Wyzwania

Pomimo łatwości implementacji i szybkiemu rozwojowi, uczenie maszynowe napotyka również szereg wyzwań, takich jak konieczność posiadania dużych, reprezentatywnych zbiorów danych oraz trudności w doborze odpowiednich algorytmów i optymalizacji ich parametrów. Przyszłość tej dziedziny będzie związana z dalszym rozwojem metod zbierania i przetwarzania potężnych ilości danych, co umożliwi uzyskiwanie bardziej precyzyjnych i trafnych prognoz. Istotnym kierunkiem rozwoju będzie także doskonalenie algorytmów, w tym tworzenie nowych, bardziej zaawansowanych metod, które będą w stanie radzić sobie z coraz bardziej złożonymi problemami, zapewniając lepszą skalowalność, wydajność oraz dokładność predykcji.

1.3. Lasy losowe

Lasy losowe (*ang. Random Forests*) zostały opracowane przez Leo Breimanna w 2001 roku jako technika uczenia maszynowego, znajdująca zastosowanie zarówno w problemach regresyjnych, jak i klasyfikacyjnych. Ich celem jest ogólna poprawa precyzji w porównaniu do pojedynczych drzew decyzyjnych podczas rozwiązywania bardziej złożonych problemów. Metoda *Random Forest* została opracowana w celu wykorzystania zalet drzew przy jednoczesnym ograniczeniu ich głównych wad.

Drzewa decyzyjne

Aby móc wyjaśnić działanie lasu losowego, należy najpierw zrozumieć zasadę działania drzewa decyzyjnego (*ang. Decision Tree*). Drzewo decyzyjne nie wykorzystuje mechanizmu propagacji wstecznej, co oznacza, że jego parametry nie są adaptowane w trakcie procesu uczenia. Drzewa decyzyjne to struktury, dążące do precyzyjnego rozwiązania problemów klasyfikacyjnych i regresyjnych, na podstawie podziału danych na coraz bardziej jednorodne podzbiory. Jednorodność może być mierzona np. za pomocą entropii, która opisuje stopień losowości w rozkładzie danych — im niższa entropia, tym bardziej jednorodny podzbiór. Każdy węzeł w drzewie odpowiada za rozdzielenie danych na podstawie jednej cechy, której określona wartość najlepiej rozgranicza jednorodne próbki od reszty. Proces ten podstawowo kontynuowany jest aż do momentu, w którym podzbiory staną się całkowicie jednorodne (można to jednak zmienić za pomocą szczególnych hiperparametrów). Ostatnie węzły stają się liśćmi drzewa, zawierając ostateczne klasyfikacje lub wartości (w przypadku regresji). Dzięki prostocie interpretacji i możliwościom łatwego modelowania, drzewa decyzyjne są popularnym narzędziem w uczeniu maszynowym.

Lasy losowe

Pomimo przejrzystości drzew decyzyjnych i łatwości w ich implementacji, wykazują one jedną podstawową wadę. Charakterystyczna jest ich duża podatność na przeuczanie (*ang. overfitting*), szczególnie w przypadku bardzo złożonych, wielowymiarowych danych. Oznacza to, że w procesie budowania drzewa mogą powstawać silnie skomplikowane, rozłożyste struktury, które doskonale dopasowują się do danych treningowych, jednak nie generalizują dobrze problemu. Powoduje to najczęściej nieprecyzyjne wyniki na zbiorach testowych.

Lasy losowe rozwiązują jednak powyższy problem. Algorytm polega bowiem na tworzeniu struktury składającej się z wielu drzew decyzyjnych, z których każde jest budowane na podstawie losowego podzbioru danych treningowych oraz losowego podzbioru cech. Ta technika jest powszechnie znana jako *bagging*. Pomaga to zmniejszyć wariancję i pozwala zminimalizować szanse na przeuczenie modelu. Następnie jest on ewaluowany na zbiorze danych, który nie został uwzględniony podczas trenowania poszczególnych drzew, nazywanym (*out-of-bag dataset*), dzięki czemu uzyskać można niezależną ocenę precyzji modelu [4]. W związku z mnogą liczbą drzew, ostateczna predykcja jest uzyskiwana poprzez uśrednienie wszystkich prognoz drzew w przypadku regresji lub głosowanie większościowe w przypadku klasyfikacji [5]. Dzięki temu las losowy lepiej uogólnia rozwiązywany problem, pomagając przez to zwiększyć precyzję modelu i ograniczyć szansę na (*ang. overfitting*).

Jak podkreślono w artykule Briemanna, lasy losowe znajdują zastosowanie w przypadku danych o dużej liczbie zmiennych wejściowych, wysokiej złożoności oraz obecności szumu i wartości odstających. Dzięki losowemu wyborowi cech oraz podziałowi na wiele drzew, algorytm *Random Forest* potrafi efektywnie radzić sobie z danymi, w których żadna pojedyncza zmienna nie pozwala na jednoznaczne przewidywanie wartości lub klasyfikacji. Dodatkowo, oferuje on wbudowane mechanizmy oceny ważności zmiennych, co ułatwia zrozumienie wpływu poszczególnych cech na wynik, a konstrukcja pozwala na szybkie obliczenia i łatwą paralelizację. Główną wadą jest jednak trudność w interpretacji wyników, lasy losowe traktowane są jako modele „*czarnej skrzynki*” [5].

1.4. Przegląd literatury

Powstało wiele projektów i artykułów naukowych dotyczących problemu wybrania najbardziej wartościowego gracza lub laureata innych plebiscytów sportowych. Jednak większość z nich prezentuje odmienne podejście lub zakłada inny cel niż przedstawione w pracy. Wykorzystanie w nich różnego rodzaju algorytmów uczenia maszynowego, czy odmiennych sposobów przygotowania danych wpływa bezpośrednio nie tylko na ostateczny rezultat, ale także na przejrzystość wyników i zrozumienie sedna problemu.

W artykule pod tytułem „*Forecasting the NBA's Most Valuable Player: A Regression Analysis Approach*" [6], autorstwa Arnando Harlianto i Johan'a Setiawan'a, zaprezentowane zostało rozwiązanie problemu za pomocą trzech algorytmów: regresji liniowej, drzewa decyzyjnego oraz SVR (Support Vector Regression), z których to regresja liniowa okazała się najskuteczniejsza. W trakcie przygotowywania danych tworzona jest sztuczna zmienna Fantasy Points, bezpośrednio wynikająca z kilku kluczowych statystyk każdego z graczy. Ma ona pełnić rolę wszechstronnej metryki, a jednocześnie zmiennej będącej predykowaną wartością - po posortowaniu uzyskuje się prognozowanego zwycięzcę nagrody MVP. Odmienność tego konkretnego podejścia wyróżnia fakt, iż nie tylko zdecydowano o ścisłej grupie oficjalnych statystyk używanych do spreparowania zmiennej, ale także ręcznie przypisano wagę każdej z nich. Ma to na pewno bezpośredni wpływ na precyzję algorytmów. Hipoteza zakłada, że gracze z wysoką wartością Fantasy Points powinni być silnymi kandydatami do wygrania plebiscytu.

Artykuł autorów Mason'a i Charles'a Chen pod tytułem „Data Mining Computing of Predicting NBA 2019-2020 Regular Season MVP Winner" [7] przedstawia podejście bardziej analityczne, koncentrując się na statystyce i metodach Big Data. Obliczane są różnego rodzaju indeksy MVP, które mają agregować wcześniej przebrane wyniki koszykarza w jedną, indywidualną wartość dla danego gracza. Tworzenie indeksów ma odzwierciedlać kolejne poziomy głębokości analizy osiągnięć gracza oraz jego drużyny. Cennym wyróżnikiem jest również redukcja wymiarowości, osiągana dzięki budowaniu statystycznych indeksów. Ostatecznie stosowana jest sztuczna sieć neuronowa, która wykorzystuje istniejące indeksy do prognozy prawdopodobieństwa na zostanie MVP.

W publikacji Gabriela Pastello - „Predicting the NBA MVP with Machine Learning" [2] sugerowane jest, że najbardziej wartościowy gracz niekoniecznie musi być tym najlepszym z punktu widzenia statystyki. Aby zredukować jednak liczbę pretendentów na sezon i odfiltrować graczy z szansami bliskimi zeru, ustalane są minimalne kryteria, które gracz musi spełnić, aby być branym pod uwagę w predykcji. Wartości tych kryteriów nie są określane bezpośrednio przez autora, lecz opierają się na ich historycznie minimalnych wartościach dla graczy, którzy zdobyli tę statuetkę. W pracy testowany jest szereg modeli uczenia maszynowego, a wydajność każdego z nich oceniana jest za pomocą tych samych miar: *RMSE* (błędu średniokwadratowego) oraz R^2 (współczynnika determinacji), żeby móc je porównać. Artykuł przedstawia najbardziej intuicyjne podejście do rozwiązania problemu prognozowania laureata nagrody MVP, jednak nie wszystkie testowane algorytmy radzą sobie jednak z precyzyjnym rozwiązaniem postawionego problemu.

1.5. Środowisko i wykorzystane narzędzia

Po przeanalizowaniu literatury oraz rozważeniu dostępnych rozwiązań technologicznych zdecydowano się na wykorzystanie języka *Python*, jako głównego narzędzia. Jak wynika z powyższych publikacji, a także z doświadczeń praktycznych, język ten jest niezwykle przystępny i elastyczny, co czyni go szczególnie popularnym w obszarze tworzenia i rozwijania modeli uczenia maszynowego. *Python* oferuje bogaty zbiór bibliotek, takich jak *Pandas*, *NumPy*, *scikit-learn* czy *keras*. Wspierają one proces zbierania, czyszczenia i obróbki danych, pozwalają na implementację oraz testowanie modeli uczenia maszynowego a także oferują wiele narzędzi statystycznych do jego poźniejszej walidacji oraz wizualizacji wyników. Dodatkowym atutem *Pythona* jest potężna społeczność, a zatem i rozbudowana dokumentacja powyższych i wielu innych bibliotek. Dzięki temu możliwe jest szybkie i skuteczne tworzenie i edycja zbiorów danych, budowanie struktur modeli oraz trenowanie ich od podstaw, co znacząco przyspiesza proces badawczy i ma ułatwić rozwiązanie problemu zadanego w tytule projektu.

Dokonano także wyboru środowiska umożliwiającego pracę w języku *Python*. W procesie wyboru istotną rolę grała dostępność środowiska z poziomu wielu urządzeń, co miało pozwolić na korzystanie z najnowszych wersji kodu niezależnie od miejsca pracy. Ważnymi aspektami przy wyborze była prostota konfiguracji oraz intuicyjność obsługi, a zarazem możliwość skorzystania z szerokiej gamy dostępnych bibliotek *Pythona*, zwłaszcza tych związanych z uczeniem maszynowym. Z tych względów zdecydowano się na rozpoczęcie projektu w środowisku Kaggle[2].

Kaggle to internetowa społeczność oraz platforma zrzeszająca naukowców, inżynierów i entuzjastów zajmujących się danymi oraz inżynierią uczenia maszynowego. Opiera się na *Jupyter Notebook*, który jest jednym z najpopularniejszych narzędzi do tworzenia projektów w języku *Python* online. Pozwala tworzyć notatniki do implementacji kodu, który potem wykonuje się na maszynach wirtualnych i pokazuje wyniki na stronie internetowej. Dzięki temu, projekt można tworzyć na wielu urządzeniach, niezależnie od miejsca pobytu. Kluczowym aspektem wyboru Kaggle ponad np. popularniejsze *Google Collab* była wygodna implementacja systemu kontroli wersji, umożliwiająca śledzenie postępów i cofanie się do wcześniejszych wersji kodu w miarę rozwoju projektu. Ponadto, na decyzję wpływ miała również bardziej przejrzysta i intuicyjna struktura interfejsu, a także łatwy dostęp do licznych bibliotek uczenia maszynowego, które nie wymagają wcześniejszej instalacji przez użytkownika. Platforma oferuje także możliwość tworzenia własnych zbiorów danych i modeli oraz ich bezpieczne przechowywanie oraz udostępnianie, co daje możliwość korzystania z nich w różnych projektach, co pozwala podzielić pracę na wiele osobnych notatników.

[2]Kaggle - platforma wykorzystana w projekcie

2. Implementacja

W rozdziale przedstawiono proces implementacji narzędzia predykcyjnego, umożliwiającego prognozowanie MVP w dowolnie wybranym sezonie. Omówiono kolejne etapy, począwszy od zebrania i przygotowania zestawu danych. Następnie opisano proces tworzenia modeli Regresji Lasu Losowego i wykorzystania ich do predykcji MVP. Następnie opisano proces analizy wyników uzyskanych w wyniku uczenia modeli oraz wskazano wprowadzone poprawki i adaptacje, mające na celu zwiększenie ich skuteczności w rozwiązywaniu założonego problemu. W końcowej części rozdziału zaprezentowano wyniki końcowe modelu ostatecznego.

2.1. Zbiór danych

Zbiór danych to w kontekście uczenia maszynowego zestaw uporządkowanych informacji, które są wykorzystywane do trenowania, walidacji i testowania modeli i algorytmów. Dane mogą przyjmować różne formy - tabelaryczne, liczbowe, obrazów, tekstów czy nawet dźwięków. Taki zbiór składa się z przykładów (rekordów/próbek/wierszy), gdzie każdy z nich ma przypisaną grupę cech (atrybutów/parametrów) opisujących te dane. W przypadku zadań nadzorowanych, dodatkowo konieczne jest zebranie odpowiednich wartości docelowych, inaczej zwanych etykietami, które model będzie uczyć się przewidywać, co zostało przeze mnie opisane w sekcji 1.2. Jeśli jest to niemożliwe lub dane są niepełne, powinno się je uzupełnić ręcznie lub używając odpowiednich do tego narzędzi. Zbiór danych tworzy nieodłączną podstawę do budowania modelu uczenia maszynowego, a jego jakość, rozmiar i reprezentatywność mają bezpośredni wpływ na skuteczność, łatwość skalowania i udoskonalania modelu.

Zbiór danych wykorzystany w projekcie powstał na podstawie danych pozyskanych ze strony Basketball Reference [8]. Serwis ten udostępnia liczne zestawienia, które zawierają wszystkie oficjalne statystyki z NBA na dany sezon, zaczynając od 1946 roku. Mowa tutaj nie tylko o statystykach indywidualnych, a także tabelach rankingowych po sezonie, które odzwierciedlają miejsca zajmowane przez drużyny po jego zakończeniu. Basketball Reference cieszy się reputacją jednego z najbardziej wiarygodnych źródeł gromadzących tego typu dane. Ze względu na to stanowi doskonałą bazę do stworzenia zbioru potrzebnego do realizacji projektu.

Dane zostały zebrane w formie tabelarycznej, gdzie pojedynczym rekordem jest zbiór statystyk danego koszykarza, oznaczony jego imieniem i nazwiskiem. Stworzenie pełnego zestawienia statystyk gracza na dany sezon, jest możliwe przez połączenie 4 osobnych tabel, pobranych ze strony Basketball Reference [8]. Zostały one wymienione poniżej:

- **Tabela średnich statystyk na mecz** — w której wartość każdej statystyki dzielona jest przez liczbę rozegranych meczów przez zawodnika.
- **Tabela statystyk zaawansowanych** — obejmującą szczegółowe wskaźniki, mające wspomóc dopasowanie modelu do problemu.
- **Tabela statystyk sumarycznych** — zawierającą sumaryczne wartości wszystkich statystyk zebranych przez koszykarza w danym sezonie regularnym.
- **Tabela klasyfikacji drużyn** - przedstawiającą ostateczną klasyfikację drużyn po zakończeniu danego sezonu regularnego

Aby model mógł skutecznie uogólnić problem, kluczowe jest dysponowanie jak największym zbiorem danych. Z uwagi na coroczną selekcję MVP, konieczne było zgromadzenie i utworzenie zestawień sezonowych dla wielu lat wstecz. Stworzono 23 takie zestawienia, łącząc w każdym z nich 4 wyżej wymienione tabele. Każde zestawienie odpowiada pełnemu zbiorowi statystyk z pojedynczego sezonu regularnego NBA (od 2001 do 2023). Skupienie się na najnowszych sezonach wynika z dążenia do przewidywania przyszłych laureatów nagrody MVP. Statystyki graczy sprzed 2000 roku mogłyby zawierać nieaktualne wzorce, odzwierciedlające zmieniający się poziom indywidualnych umiejętności zawodników na przestrzeni lat.

W efekcie utworzono ogólną strukturę zbioru danych, opierającą się o połączenie 3 z 4 przygotowanych wcześniej tabel - wykluczając na razie tabelę klasyfikacji drużyn. Połączenie zostało wykonane na podstawie imienia i nazwiska gracza, w danym sezonie. Z tego powodu, zawodnicy figurujący w tabeli sezonowej kilkukrotnie, ze względu na transfery, zostali podsumowani przez Basketball Reference do jednej sumarycznej próbki, która w kolumnie 'tm' - drużyna, zawiera wartość 'TOT'. Duplikaty wierszy z tą samą kolumną 'player' zostały następnie usunięte, pomijając wiersze z kolumną 'tm' o wartości 'TOT'. Zostawiono zatem ostatecznie po jednym wierszu reprezentującym każdego aktywnego gracza w danym sezonie. Należy zauważyć, że z wyjątkiem podsumowania statystyk gracza transferowanego i usunięcia jego duplikatów spowodowanych transferami, zbiór danych zawiera statystyki i próbki w ich pierwotnej formie, przed jakimkolwiek wstępnym przetwarzaniem. W tabeli 2.1 przedstawiono ogólną strukturę zbioru danych z przykładami, a wymienione w tytułach kolumn grupy statystyk są szczegółowo opisane w Dodatku 1.

Zawodnik	Drużyna	Średnie statystyki na mecz	Statystyki zaawansowane	Statystyki sumaryczne
Lebron James	LAL	...	...	...
Luka Dončić	DAL	...	...	...
Bobby Portis	MIL	...	...	...

Tabela 2.1. Ogólna struktura zbioru danych z przykładami

W celu uporządkowania zbioru danych, nazwy kolumn z tabeli średnich statystyk na mecz zostały rozszerzone o sufiks *per_game*, natomiast kolumn z tabeli statystyk sumarycznych o sufiks *total*. Nazwy kolumn statystyk zaawansowanych pozostały unikalne, dlatego żaden sufiks nie został do nich dodany. Następnie wykonano następujące operacje:

1. wszystkie nazwy parametrów zapisano małymi literami dla większej czytelności,
2. ze względu na połączenie tabel pokrywających się częściowo nazwami kolumn, należało usunąć ich duplikaty i bezsensowne formy powstałe po dodaniu sufiksów. Z tego względu:
   - usunięto kolumny *rk*, *rk_per_game* oraz *rk_total* odpowiadające za id zawodnika na stronie Basketball Reference. Do trenowania modelu wymienione kolumny nie są konieczne,
   - kolumnę *gs* oznaczającą liczbę gier, w których zawodnik rozpoczął mecz w pierwszym składzie w całym sezonie, zapisano jako *gs_total* dla czytelności i powtarzalności tabeli. Następnie usunięto jej duplikat z tabeli średnich statystyk na mecz - *gs_per_game*, który po dodaniu sufiksu nie ma sensu,
   - usunięto także kolumnę *position*. W dzisiejszych czasach pozycja zawodnika odgrywa coraz mniejszą rolę przy głosowaniu na MVP. Pozostawienie tej cechy mogłoby wpływać negatywnie na precyzję modelu w przyszłości.
3. zmieniono nazwę kolumny *tm* na *team* dla zwiększenia czytelności,
4. usunięto kolumnę *player-additional*. Została ona wprowadzona przez stronę Basketball Reference żeby zastąpić kolumnę *rk*, dopiero w sezonach 2023 i 2024, co spowodowało utworzenie wielu pustych komórek podczas dołączania zestawień do zbioru danych. Podobnie jak w przypadku *rk* ten parametr również nie jest potrzebny do uczenia modelu,
5. utworzono słownik zawierający skróty nazw drużyn przypisane do ich pełnych nazw (np. *DAL - Dallas Mavericks*). Następnie za jego pomocą podmieniono wartości kolumny *team* na skróty nazw drużyn, zastępując ich pełne nazwy. Ich obecność w zbiorze danych jest kluczowa dla kolejnego podpunktu,
6. uwzględniając wcześniej podkreśloną wagę wpływu gracza na wyniki drużyny, utworzono nową kolumnę *seed*, zawierającą miejsce drużyny w tabeli sezonowej, wewnątrz tabeli klasyfikacji drużyn. Kolumna powstała poprzez wykorzystanie istniejących już kolumn *W* - ilość wygranych meczów oraz *L* - ilość przegranych meczów przez drużynę. Na ich podstawie podsumowano bilanse i posortowano klasyfikację w każdym roku. Zespół

z najlepszym bilansem otrzymał zatem wartość *seed* równą 1, a kolejne 29 drużyn odpowiednie wartości do ich miejsca w rankingu. Ze względu na równy podział 30 drużyn NBA na konferencje wschodnią i zachodnią, najgorsze miejsce w tabeli to miejsce 15,

  – wykorzystując wspólną już kolumnę *team* docelowego zbioru danych oraz posortowanej tabeli klasyfikacji drużyn, każdemu graczowi nietransferowanemu w ciągu sezonu została przypisana wartość *seed* jego drużyny. Możliwe to było tylko dzięki wcześniejszej podmianie pełnych nazw drużyn na odpowiadające im skróty,

  – gracze transferowani zostali natomiast potraktowani inaczej. Przez wcześniejsze podsumowanie ich wierszy do wspólnej próbki o wartości kolumny *team* równej *TOT*, ich wartość w kolumnie *seed* pozostała pusta - nie istnieje bowiem drużyna o skrócie nazwy *TOT*, jest tylko tylko wyróżnik graczy transferowanych. Postanowiono zatem przypisać im z góry narzuconą wartość kolumny *seed* równą 8, czyli średnią ligową. Operacja ta jest celowa - hstorycznie żaden wartościowy zawodnik mający szanse na wygranie plebiscytu MVP, nie został wytransferowany w trakcie sezonu. Nie będzie on zatem kluczową próbką danych. Uśredniając jednak wartość jego parametru *seed* możemy wykorzystać wiele próbek, dotyczących graczy transferowanych, do uczenia modelu,

7. dodano kolumnę *season*, która jednoznacznie wyróżnia, w którym sezonie gracz zebrał określone w wierszu statystyki. Jest to kluczowe dla późniejszego porównania zawodników i wybrania MVP spośród kandydatów z pojedynczego sezonu,

8. ostatecznie, stworzona i dodana została kolumna z wartością prognozowaną - *mvp votes share*. Z uwagi na zmieniającą się liczbę ekspertów z roku na rok, wykorzystanie samej liczby głosów, dostępnej na oficjalnej stronie NBA, jako wartości prognozowanej przez model mogłoby niestety prowadzić do spadku jego precyzji. W związku z tym wykorzystano statystykę, która w jednoznaczny sposób wskazuje na wielkość udziału głosów na zawodnika we wszystkich możliwych głosach na MVP w danym sezonie, niezależnie od liczby głosujących. W pracy Gabriela Pastorello wyprowadzono przystępną formę tej statystyki [2], przedstawioną za pomocą wzoru (2.1.1).

$$MVS = \frac{PV}{TGV} \tag{2.1.1}$$

Opis zmiennych:

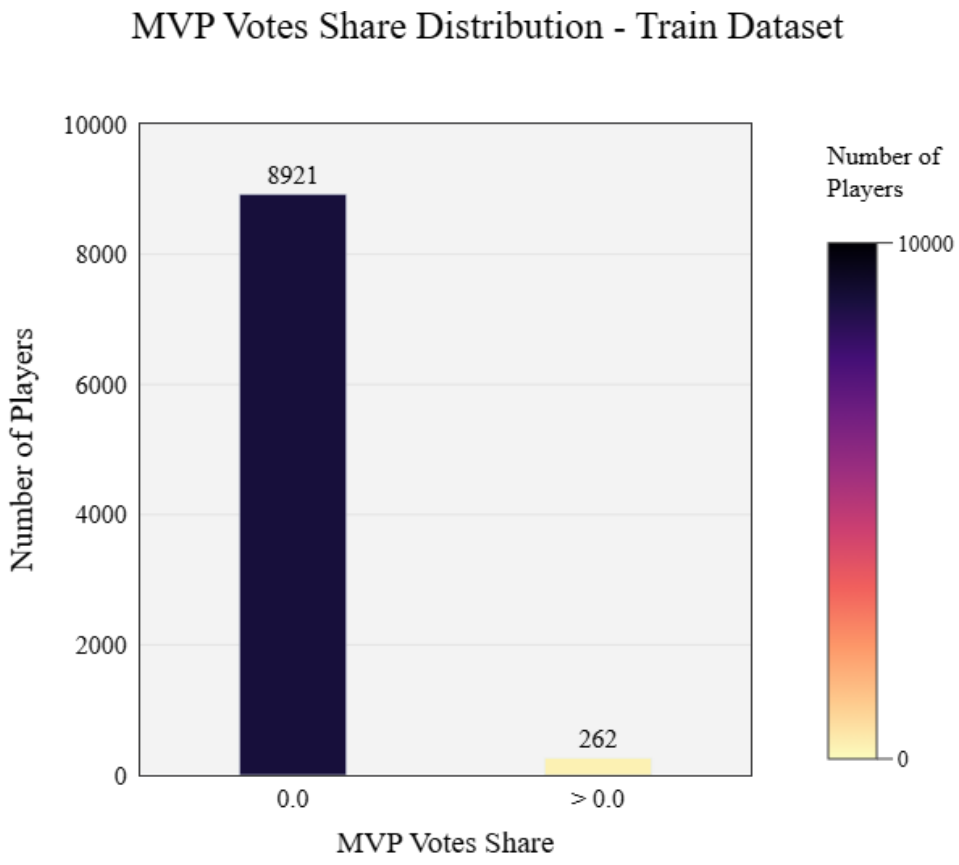
- MVS (*MVP Votes Share*) - udział głosów oddanych na gracza w całym głosowaniu,
- PV (*Player Votes*) – liczba głosów oddanych na gracza,
- TGV (*Total Given Votes*) – liczba wszystkich możliwych głosów oddanych w głosowaniu.

Każdemu graczowi została przypisana wartość *mvp votes share* na podstawie danych głosowań z oficjalnej strony NBA. Zawodnicy nieuwzględnieni w głosowaniu otrzymali wartość kolumny *mvp votes share* równą 0. Po wykonaniu powyższych operacji, zbiór danych stanowią 23 obszerne tabele zawierające informacje ogólne, statystyki średnie na mecz, zaawansowane, sumaryczne, informację o sezonie w którym statystyki zostały zebrane oraz wartość prognozowaną - *MVP Votes Share*. Zestawienia poddano następnie podziałowi na standardowe zbiory używane w uczeniu maszynowym. Z uwagi na powszechną zasadę podziału danych w proporcji około 80% na zbiór treningowy i 20% na zbiór testowy, wybrano i połączono cztery konkretne zestawienia (sezony 2023, 2022, 2011 oraz 2003) do testowania precyzji modelu. Dokonany wybór jest zamierzony - dwie najświeższe tabele mają posłużyć do oceny modelu w warunkach kluczowych dla przyszłych zastosowań, takich jak prognozowanie wyników w następnych latach. Uwzględnienie sezonów ze środka oraz z początku wybranego okresu ma natomiast na celu sprawdzenie, czy model nie wykazuje tendencji do nadmiernego dopasowywania się do najnowszych lat. Wybór większej ilości tabel do zbioru testowego nie jest w ten sposób potrzebny, a dzięki temu więcej rekordów może być używane do treningu modelu. Pozostałe 19 tabel połączono zatem w rozbudowany zbiór treningowy. W tabeli 2.2 została przedstawiona ostateczna struktura obydwóch zbiorów z przykładami. Należy zaznaczyć iż jedyne kolumny w formie tekstowej, czyli *player* oraz *team* znajdują się w zbiorze wyłącznie dla celów związanych z wyświetlaniem wyników. Wszystkie rodzaje statystyk z poniższej tabeli są szczegółowo opisane w Dodatku 2.

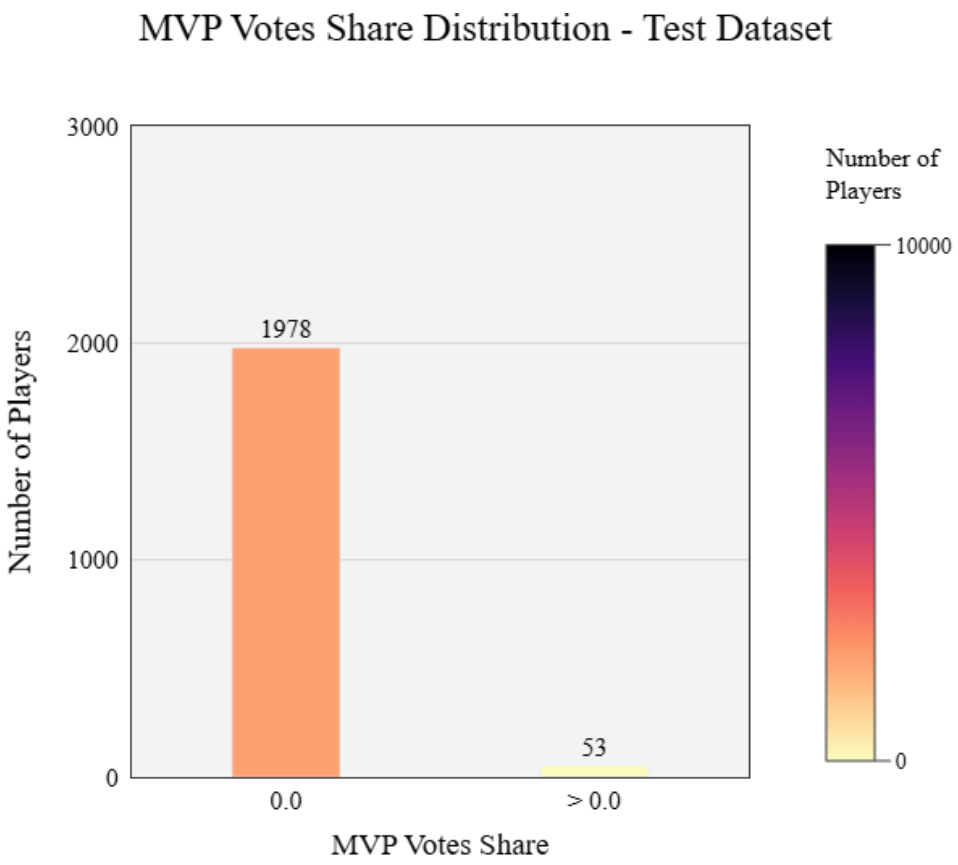
Zawodnik	Drużyna	Średnie statystyki na mecz	Statystyki zaawansowane	Statystyki sumaryczne	Sezon	MVP Votes Share
Lebron James	LAL	...	...	...	2012	0.998
Luka Dončić	DAL	...	...	...	2023	0.572
Bobby Portis	MIL	...	...	...	2023	0.0

Tabela 2.2. Ostateczna struktura zbioru danych z przykładami

Pozostała kwestia rekordów niepełnych, czyli zawierających puste wartości w niektórych kolumnach. Zdecydowano się na zastąpienie wszelkich wartości *NaN* średnią z odpowiadających im kolumn. Celem tej procedury jest dalsze wyróżnienie graczy, którzy znacząco odbiegają od średniej w statystykach. Dzięki temu zachowano jednocześnie wszystkie rekordy niepełne. Ilość danych oraz jakość zbioru treningowego są kluczowe dla osiągnięcia satysfakcjonujących rezultatów w późniejszym procesie uczenia modelu. Zbiór danych składa się 9183 próbek statystyk zawodników w części treningowej, dane służące do testowania modelu zawierają natomiast 2031 rekordów. Każda próbka posiada dokładnie 75 cech wymienionych i opisanych w Dodatku 2. Należy tutaj podkreślić ogromne niezbalansowanie zbioru danych. Na wykresach 2.1 oraz 2.2 pokazano kolejno poziom niezbalansowania zestawu treningowego i testowego. Niewielka ilość graczy mających wartość *MVP Votes Share* wyższą niż 0.0 wynika z corocznej formy głosowania, co omówiono dokładnie w rozdziale 1 w sekcji 1.1.



Rys. 2.1. Ilość zawodników w zbiorze treningowym w zależności od wartości *MVP Votes Share*



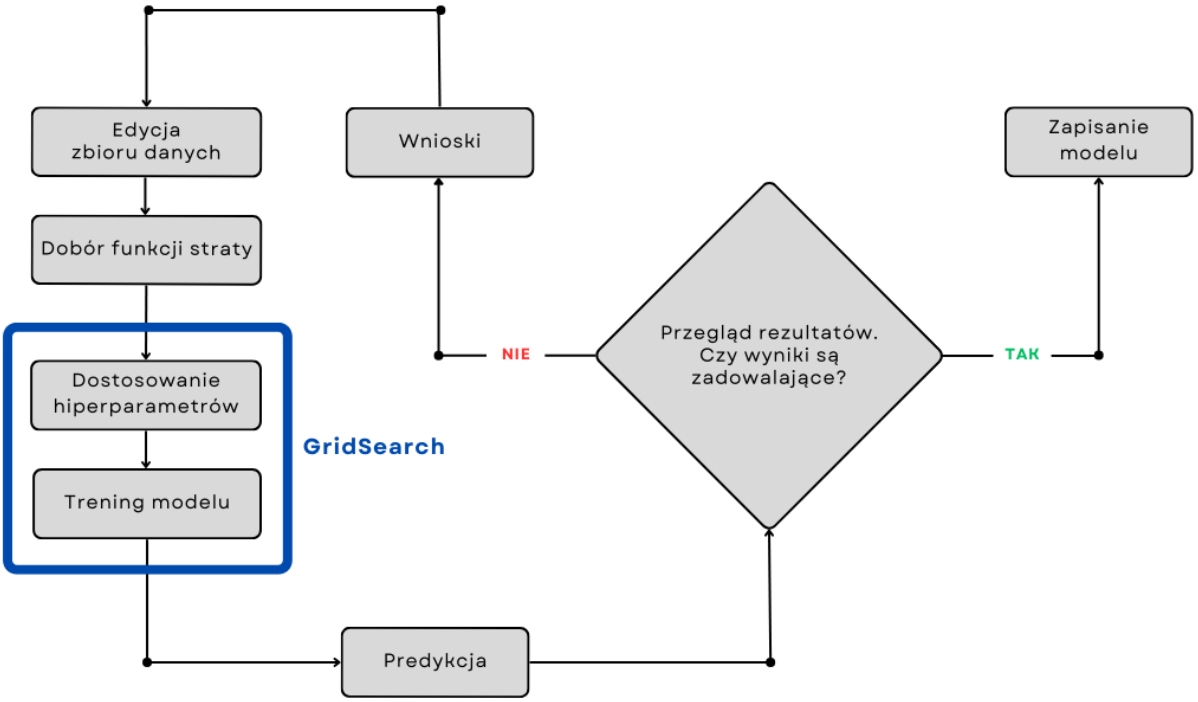
Rys. 2.2. Ilość zawodników w zbiorze testowym w zależności od wartości *MVP Votes Share*

2.2. Systematyzacja procesu uczenia

Biorąc pod uwagę poziom niezbalansowania oraz wielowymiarowości danych, zdecydowano się na użycie Lasu Losowego do rozwiązania problemu prognozowania najbardziej wartościowego gracza NBA. Opis zalet zastosowania tej metody, które spowodowały o jej ostatecznym wyborze, można znaleźć w rozdziale 1 sekcji **??**. Niewielka liczność klasy zawodników MVP wyklucza możliwość rozpatrzenia problemu metodą klasyfikacji. Faktycznych zwycięzców plebiscytu mamy bowiem 23, przy ponad dziesięcio-tysięcznej liczności grupy non-MVP. Augmentacja przy tej proporcji liczności klas mogłaby być ciężka i nie dawać zadowalających efektów. Zdecydowano się zatem na podejście regresyjne, z użyciem modelu Regresji Lasu Losowego (*ang. Random Forest Regressor*). Przeprowadzono rozeznanie w sposobach jego implementacji w języku *Python*, ostatecznie wybierając popularną bibliotekę *scikit-learn*, oferującą szeroką dokumentacje metod przygotowania, zastosowania i testowania algorytmu[1]. Przed rozpoczęciem implementacji, stworzono schemat działania przedstawiony na Rys 2.3.

[1]Biblioteka sckit-learn - dokumentacja modelu *Random Forest Regressor*

Zawiera on iteracyjny i usystematyzowany proces badań, mający pozwolić na wyciąganie wniosków z rezultatów kolejnych prób uczenia. Zachowanie kolejności oraz podział na poszczególne etapy umożliwia skuteczniejsze identyfikowanie błędów oraz precyzyjniejsze dopasowanie odpowiednich warunków i parametrów do trenowania modelu.



Rys. 2.3. Schemat działania podczas uczenia modelu

Każdy z kroków powyższego diagramu został wydzielony ze względu na swój znaczny wpływ na wyniki predykcji. Starannie zaplanowano, a następnie zaimplementowano każdy z nich przy pomocy języka *Python* w środowisku Kaggle. Implementacja miała być możliwie uniwersalna i pozwalać na ulepszanie, poszerzanie procesu procesu w razie potrzeby.

- Edycja zbioru danych i jego dostosowanie jest kluczowe dla lepszego uogólnienia problemu dla modelu. Pomimo silnej odporności Lasu Losowego na przeuczenie, wzięte zostało pod uwagę duże niezbalansowanie i wysoka złożoność danych. Niewykluczona została zatem możliwość adaptowania zbioru treningowego, na podstawie wniosków wyciągniętych z wyników kolejnych predykcji,

- Dobór/adaptacja funkcji straty pozwala natomiast odpowiednio nagradzać lub karać model w trakcie uczenia. Wybór odpowiedniej z pośród wielu dostępnych funkcji, stosowanych w implementacjach regresyjnych uczenia maszynowego, ma znaczny wpływ na trafność wyników i ostateczną precyzję modelu,
- Dostosowanie hiperparametrów i trening modelu zostały połączone za pomocą klasy *GridSearchCV*[2] biblioteki *sckit-learn*. Jest to bardzo popularne narzędzie, służące do automatycznego dostrajania hiperparametrów wybranego modelu, przeprowadzając systematyczne przeszukiwanie zdefiniowanej uprzednio siatki ich wartości (*ang. param grid*). Model jest następnie trenowany osobno dla każdej możliwej kombinacji. Z tego względu wybranie hiperparametrów do *param grid* powinno być wcześniej przemyślane. Zbyt niewielka ilość możliwości nie pozwoli zbliżyć się do optimum globalnego, natomiast jeśli wariacji będzie za dużo, znacznie wydłuży się czas szukania rozwiązania. Wybór takiego podejścia wiązał się zatem z selekcją hiperarametrów umieszczonych w *param grid*. Posłużono się dokumentacją *Random Forest Regressor*[3] aby wybrać grupę - zdaniem autora - najważniejszych z nich:
    - *n_estimators* - liczba drzew decyzyjnych w lesie losowym, wyrażona jako liczba całkowita. Jej wartość kontroluje wielkość lasu, wpływając na dokładność modelu i czas obliczeń [9],
    - *bootstrap* - parametr logiczny określający, czy próbki do budowania kolejnych drzew decyzyjnych mają być wybierane losowo z powtórzeniami. Domyślnie ustawiony na *True*, umożliwiając stosowanie metody próbkowania bootstrapowego. Przy wartości *False* do budowy każdego drzewa decyzyjnego wykorzystywany jest cały zbiór danych [9],
    - *max_samples* - wartość zmiennoprzecinkowa definiująca część próbek, którą algorytm losowo wybiera do utworzenia każdego drzewa decyzyjnego. Dostępna tylko, gdy *bootstrap* jest ustawiony na *True* [9],
    - *max_depth* - maksymalna głębokość drzewa, wyrażona jako liczba całkowita. Jeśli nie jest ustawiona, drzewa rozrastają się do pełnej czystości liści, co może prowadzić do przeuczenia [9],
    - *random_state* - liczba całkowita kontrolująca losowość metody *bootstrap*. Ustawienie stałej wartości tego parametru umożliwia uzyskiwanie powtarzalnych wyników i rzetelne porównywanie różnych wersji modelu w trakcie kolejnych iteracji [9].

[2]Biblioteka sckit-learn - dokumentacja algorytmu *GridSearchCV*
[3]Biblioteka sckit-learn - dokumentacja modelu *Random Forest Regressor*

Wybrane hiperparametry zostały uznane za wystarczające, nie wykluczając dodawania kolejnych. Skorzystano także z opcji walidacji krzyżowej, którą oferuje klasa *GridSearchCV*, aby dogłębniej ocenić skuteczność każdej osobnej konfiguracji. Powyżej opisane podejście pozwoliło na usprawnienie procesu szukania najlepszych wartości hiperparametrów dla modelu Regresora Lasu Losowego.

- Predykcje, zarówno na zbiorze treningowym jak i testowym, również wykonano za każdą iteracją procesu przedstawionego na diagramie. Dzięki temu pozytywny lub negatywny wpływ każdej zmiany w dowolnym kroku mógł być zauważony. Bez predykcji przegląd i podsumowanie rezultatów nie byłyby możliwe, a każda kolejna iteracja procesu nie dawałaby wystarczającej ilości informacji o skuteczności modelu.
- Wnioski wyciągano zatem na podstawie wyników predykcji, przy każdej kolejnej próbie. Stworzono funkcje prezentującą je w przystępnej formie, pozwalając na sprawniejszą interpretację. Wyświetlała ona początkowo 4 rodzaje wykresów:
  - Porównanie realnych wartości *MVP Votes Share* do prognozowanych przez model na zbiorze testowym, w postaci wykresu punktowego. Miało to na celu zaprezentowanie ogólnej dokładności algorytmu,
  - Średnią i maksymalną dewiację błędów w zależności od przedziału realnej wartości *MVP Votes Share*. Najważniejsza była bowiem jak największa precyzja przy wysokich wartościach,
  - Wykres rozproszenia rzeczywistych i przewidywanych wartości na prostej, mający pokazać przeuczenie (*ang. overfitting*) lub niedouczenie (*ang. underfitting*) modelu,
  - Wykres ważności cech (*ang. feature importance*), umożliwiający ocenę, czy żadna z cech nie dominuje nad pozostałymi, co mogłoby wpłynąć na nadmierne dopasowanie modelu

2.3. Pierwszy model

W pierwszej iteracji procesu dane pozostawiono w formie szczegółowo opisanej w podrozdziale 2.1, bez wprowadzania jakichkolwiek modyfikacji do ich struktury. Takie podejście miało umożliwić zaobserwowanie wpływu wysokiego niezbalansowania zbioru danych na precyzję modelu. Przystąpiono więc bezpośrednio do kolejnego kroku, czyli wyboru odpowiedniej funkcji kosztu. Rozważono najpopularniejsze funkcje stosowane w implementacjach algorytmów regresyjnych: *MAE* (*ang. Mean Absolute Error*), *MSE* (*ang. Mean Squared Error*) oraz *RMSE* (*ang. Root Mean Squared Error*). Biorąc pod uwagę wysoki procent liczności próbek

o wartości predyktowanej równej 0.0 spodziewano się, że model lepiej poradzi sobie z prognozowaniem ich wskaźnika *MVP Votes Share*, a nawet przeuczy się właśnie w tym kierunku. Ważne dla pozytywnej oceny modelu było jednak jak najdokładniejsze przewidzenie wartości *MVP Votes Share* nie najgorszych, a najlepszych graczy. Duże błędy predykcji wśród elitarnych zawodników, mogłyby spowodować prognozę kogoś niewyróżniającego się na laureata nagrody. Z tego względu konieczne było wybranie funkcji kosztu, która intensywnie karze wysokie błędy predykcji, co zawężyło wybór do funkcji *MSE* i *RMSE*, wybierając ostatecznie drugą z nich. Motywacją było uniknięcie dużych błędów predykcji, zwłaszcza w odniesieniu do rzadkich przypadków – wysokich wartości MVP Votes Share zawodników kontendujących o tytuł MVP. Dodatkowo podparto swoją decyzję możliwością bardziej intuicyjnej interpretacji wyników, o czym w swoim artykule pisze Gabriel Pastorello [2]. *RMSE* oblicza bowiem średni błąd kwadratowy między rzeczywistymi a przewidywanymi wartościami, dzięki czemu przywiązuje większą wagę do błędów o dużych odchyleniach od wartości rzeczywistych. Średni błąd jest następnie pierwiastkowany, w związku z czym wynik jest przedstawiony w tej samej skali co wartość prognozowana.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Opis zmiennych:

- n – liczba próbek w zbiorze danych,
- y_i – rzeczywista wartość i-tej próbki,
- $\hat{y}_i$ – wartość przewidywana dla i-tej próbki.

Pozostając w tej samej skali co badana zmienna, możliwa była łatwiejsza interpretacja wyników uczenia oraz sprawniejsza adaptacja modelu przy kolejnych poprawkach. Następnie, zgodnie z ustalonym schematem,*param grid* został wypełniony wartościami hiperparametrów. Uzupełniono je sugerując się intuicją, wartościami podstawowymi opisanymi w bibliotece *Random Forest Regressor*[4] oraz artykułami pochylającymi się nad podobnym tematem.

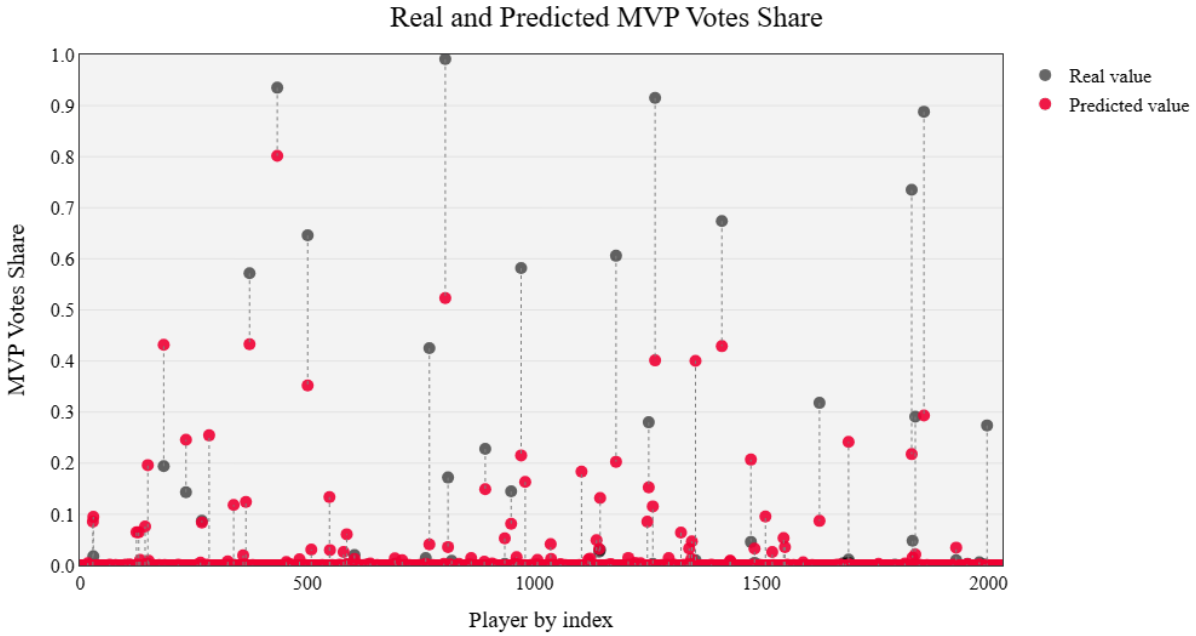
Wartości *n_estimators* objęły zatem zakres od 25 do 400, przy czym każda kolejna wartość była podwojoną wartością poprzedniej. Wynika to z podstawowej wartości równej 100, zapisanej w dokumentacji. Hiperparametr *max_samples* przyjął wartości od 0.1 do 0.9 z krokiem równym 0.1, wypełniając w zasadzie w ten sposób cały przedział, ale nie rozdrabniając go na zbyt

[4]Biblioteka sckit-learn - dokumentacja modelu *Random Forest Regressor*

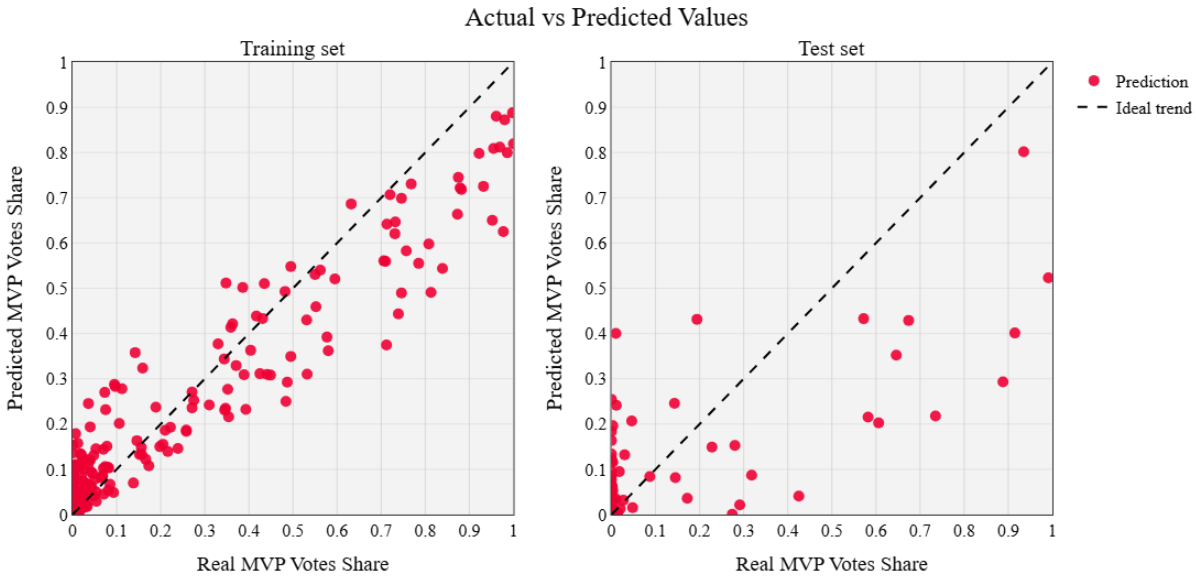
wiele możliwości do wyuczenia. Wartości *max_depth* zostały dobrane na podstawie zaleceń zawartych w artykule Sandeepa Rama pt. „Mastering Random Forests: A comprehensive guide", który sugerował stosowanie umiarkowanych głębokości drzew w celu uniknięcia *overfittingu*. Autor używa wartości 3, 5 oraz 7 [10], a w ramach tego projektu dodatkowo rozszerzono je o 9, 13, 19 i 23. Dzięki zastosowaniu narzędzia *GridSearchCV*, nie trzeba bowiem aż tak obawiać się przeuczenia, a raczej przeszukiwać w miarę możliwości jak najszerszą przestrzeń wartości hiperparametrów. *Bootstrap* został ustawiony na jego 2 jedyne wartości *True* i *False*, umożliwiając jednocześnie wybór całego zbioru danych do podziału na kolejne drzewa decyzyjne, jak i formę podziału drzew stosując *bootstraping*. Finalnie stworzono *param grid*, zawierający dostatecznie obszerne przedziały wartości, oferując jednocześnie sensowny czas wyuczenia modelu dla wszystkich kombinacji - około 40 minut. Celem pierwszej iteracji uczenia była obserwacja wyboru najlepszego zestawu parametrów, oraz dostosowania ich przedziałów przed następnym uczeniem. Gdy którykolwiek z hiperparametrów okazał się działać najlepiej przy granicznej wartości dobranego przedziału, powiększano go, a następnie uczono modele na nowo. Proces ten pozwolił na znalezienie najlepszego na tamten moment estymatora, o następujących wartościach hiperparametrów:

- *n_estimators* : 50
- *max_depth* : 23
- *max_samples* : 0.8
- *bootstrap* : True

Zastosowanie *bootstrapingu* przyczyniło się do poprawy precyzji modelu w stosunku do jego braku, co było zgodne z oczekiwaniami. Niepokojąca wydała się jednak stosunkowo wysoka wartość maksymalnej głębokości drzew decyzyjnych, sugerująca przeuczenie modelu. Wygenerowano zatem wykresy opisane szczegółowo w sekcji 2.2, oraz przeanalizowano wartości *RMSE* dla zbiorów treningowego i testowego. Wyniki wynosiły odpowiednio 0.019 dla zbioru treningowego oraz 0.028 dla zbioru testowego. Tak niskie wartości *RMSE* wskazywały wstępnie na niewielkie błędy modelu i dobre dopasowanie do danych.



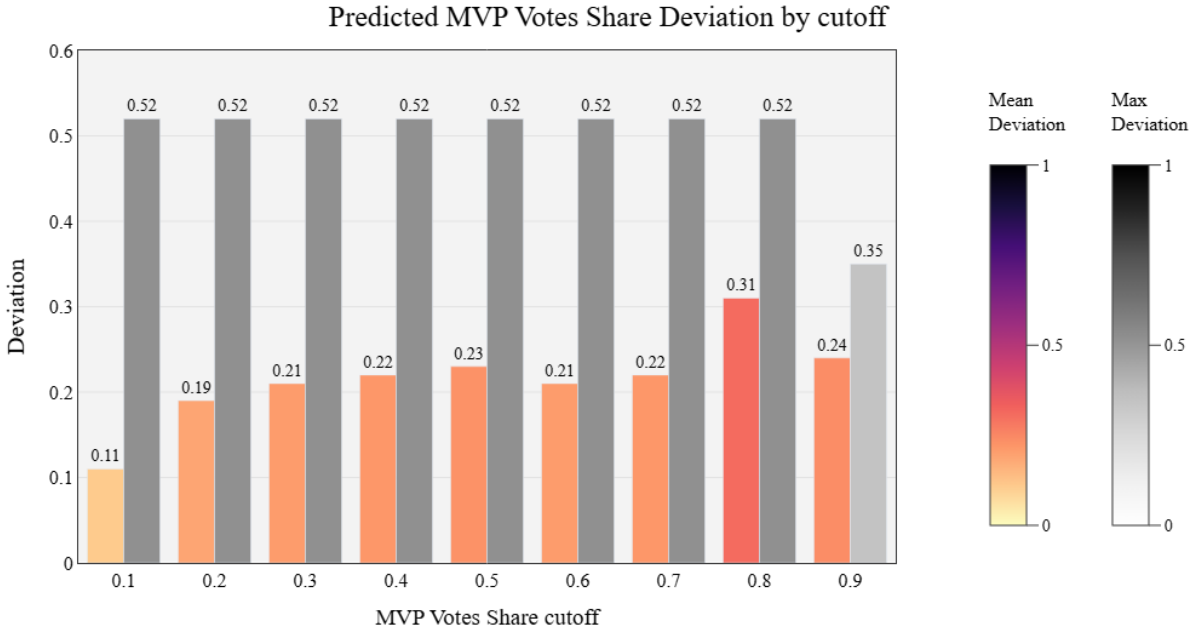
Rys. 2.4. Porównanie realnych i prognozowanych przez model 1 wartości MVP Votes Share na zbiorze testowym



Rys. 2.5. Porównanie realnych i prognozowanych przez model 1 wartości MVP Votes Share na zbiorach treningowym i testowym

Na wykresach 2.4 i 2.5 wyraźnie widać jednak skłonność do poprawnej prognozy niewielkich lub zerowych wartości *MVP Votes Share*, w przeciwieństwie do tych najwyższych. Pokazują to dobrze czarne oraz czerwone punkty ułożone w gęstą linię przy wartości 0.0 osi pionowej na wykresie 2.4.

Im wyższa wartość MVP Votes Share, tym bardziej rośnie średnia błędów modelu. Na wykresie 2.5, zbytnie dopasowanie w kierunku niskich wartości predyktora pokazane jest poprzez gęste skupienie punktów w lewym dolnym rogu. Takie zachowanie było nieoczekiwane i właśnie jego chciano uniknąć. Teorię potwierdza ostatecznie wykres 2.6 przedstawiający dewiację błędów w zależności od *MVP Votes Share*. Widoczny jest na nim wyraźny wzrost średniego odchylenia od wartości rzeczywistych w miarę zbliżania się do wartości *MVP Votes Share równej* 1.0, mimo znacznego zmniejszenia się wartości błędy maksymalnego w najbardziej interesującym przedziale.



Rys. 2.6. Wykres średnich i maksymalnych dewiacji pomiędzy predykcjami modelu 1 a wartościami realnymi

Spoglądając wagi parametrów zbioru danych dostępnych na 2.7, zauważono bardzo niepożądane mocne dopasowanie się modelu do jednego z nich. Cecha *VORP (ang. Value Over Replacement)* zdecydowanie zdominowała całą resztę. Duża waga *VORP* nie powinna dziwić, jest to złożona statystyka, pokazująca jak bardzo statystycznie lepszy jest zawodnik od swojego zmiennika. Gra zatem bardzo ważną rolę w pokazaniu wartości koszykarza na boisku. Nie powinna ona jednak praktycznie jako jedyna decydować o tym, czy ostatecznie gracz zostanie laureatem nagrody *MVP*. Początkowe wyniki modelu mogły sprawiać wrażenie obiecujących, co wynikało z silnego niezbalansowania zbioru danych. Ze względu na dużą liczebność zawodników o niskich wartościach predyktora, średni błąd modelu przyjął ostatecznie niewielką wartość.

Rys. 2.7. Wykres wagi cech zbioru danych dla modelu

Dokładniejsza analiza wykresów pozwoliła jednak na wnikliwe zbadanie precyzji modelu, co ujawniło jego rzeczywiste ograniczenia. Wyciągnięto zatem wnioski przed próbą uczenia kolejnego modelu. Ze względu na dominację niższych wartości, model dostosował się głównie do powszechnych przypadków, aby zmniejszyć wartość błędów. Zdecydowano się zatem okroić zbiór danych z graczy, którzy definitywnie nie powinni mieć szansy na wygraną w plebiscycie *MVP*. Wysokie dopasowanie modelu do cechy *VORP* skłoniło również do rozważenia dodania dodatkowej funkcjonalności, która umożliwiłaby modelowi wybór optymalnego zestawu cech z możliwością wykorzystania dowolnej ich liczby.

2.4. Adaptacja i trening drugiego modelu

Kolejną iterację procesu odnajdywania najlepszych warunków do uczenia modelu, rozpoczęto zatem od adaptacji do wniosków wyciągniętych z iteracji poprzedniej. Po pierwsze należało rozwiązać problem braku balansu w zbiorze danych. Możliwe były dwa przeciwstawne podejścia – nadpróbkowanie danych poprzez augmentację oraz usunięcie określonej liczby próbek z najliczniejszej grupy zawodników. Augmentacja polegałaby na sztucznym zwiększeniu liczby rekordów należących do grupy mniejszościowej (czyli zawodników o wysokim *MVP Votes Share*) poprzez modyfikację istniejących obserwacji, stosując operacje takie jak skalowanie, uśrednianie i dodawanie zakłóceń do wartości ich statystyk. Opcja ta została odrzucona, ze względu na specyfikę głosowania w procesie wyboru MVP. Stworzenie dodatkowych zawodników, o jakiejkolwiek wartości predyktora wyższej od zera, zmusiłoby również do zmian wartości *MVP Votes Share* zawodników już istniejących. Wartość *MVP Votes Share* jest bowiem obliczana ze wzoru, biorącego pod uwagę głosy na wszystkich zawodników, tak jak opisano w sekcji 2.1.

Postanowiono zatem zredukować liczność grupy zawodników posiadającej wartość *MVP Votes Share* na poziomie równym 0.0. Nie było możliwości całkowitego usunięcia tej grupy, wtedy bowiem model mógłby przeuczyć się w drugą stronę. Należało zatem pozostawić grupę zawodników wykazujących wysoki poziom umiejętności, pomimo zerowej wartości *MVP Votes Share*. Zawodnik prezentujący dobre statystyki indywidualne mógł nie znaleźć się bowiem w rankingu najbardziej wartościowych zawodników, z powodu ogólnie wysokiego poziomu rywalizacji w danym sezonie. W innym sezonie mógłby już być za to wskazywany jako kandydat do wygrania plebiscytu. Opracowano zatem formę filtrowania progowego *(ang. tresholding*), która pozwalała na usunięcie próbek graczy o teoretycznie najmniejszej lub nawet zerowej szansie na zdobycie nagrody *MVP*. Wybrano 8 ważnych i prostych do interpretacji statystyk, a następnie ustalono wartość progu dla każdej z nich. Jeżeli wartość danej cechy/statystyki zawodnika znajdowała się pod progiem oraz miał on wartość *MVP Votes Share* równą 0.0, odrzucano jego

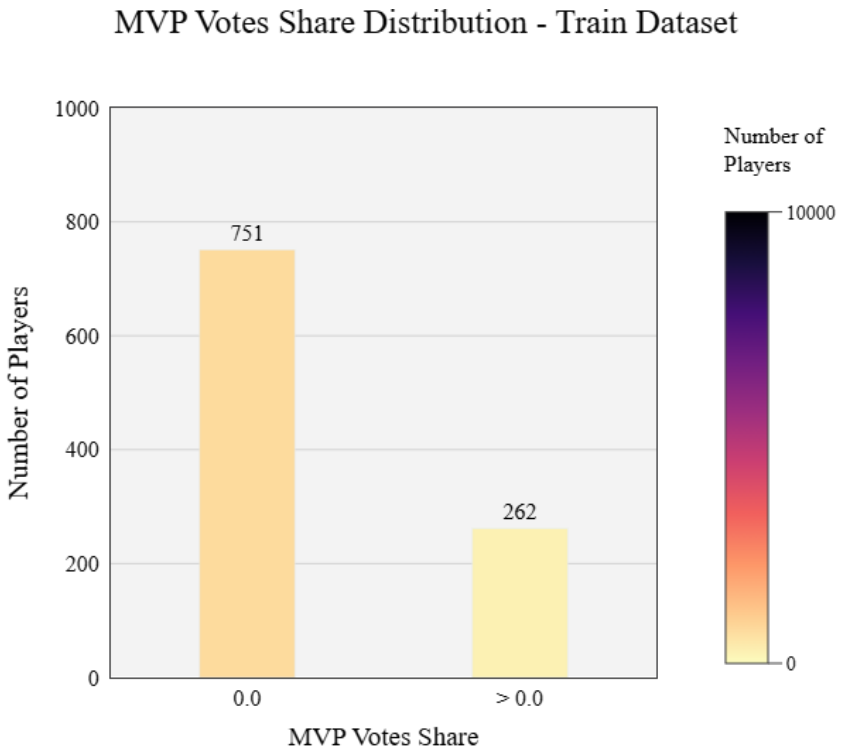
próbkę ze zbioru treningowego. Dzięki temu wszyscy koszykarze o niezerowej wartości predyktora pozostali w zbiorze danych, a z grupy zawodników bez głosów, pozostała jedynie statystycznie elitarna część. Wybór atrybutów oraz wyjaśnienie ich wartości progowych wyjaśniono poniżej:

- wszyscy zawodnicy o wartości *team = TOT* (zawodnicy wytransferowani w trakcie sezonu) zostali usunięci ze zbioru treningowego. Historycznie żaden zwycięzca plebiscytu MVP nie został wymieniony w trakcie sezonu, w którym wygrał nagrodę. Zasada ta była jak najbardziej sensowna, żadna drużyna nie powinna bowiem chcieć wymienić gracza na poziomie MVP, a raczej wykorzystać jego potencjał do wygrania mistrzostw. Ten próg pozwolił na usunięcie 1115 próbek,
- w sezonie 2023-2024 wprowadzono nową zasadę, która miała ograniczyć ilość zawodników, na których można głosować w ramach plebiscytu. Zawodnik musiał od tego czasu zagrać minimum 65 meczów w danym sezonie, aby móc być branym pod uwagę w głosowaniu. Ze względu na to, że tak wysoki próg powodował redukcję aż 4521 próbek, a także przez dosyć niedawne wprowadzenie powyższej zasady, zdecydowano się obniżyć próg do minimum 50 zagranych meczów, usuwając w ten sposób 3049 rekordów,
- najbardziej intuicyjną i jedną z najważniejszych statystyk indywidualnych zawodnika, jest średnia liczba zdobytych punktów na mecz (*pts_per_game*). Najmniej punktującym w historii koszykarzem, który zdobył nagrodę *MVP* jest Wes Unseld, trafiając średnio 13.8 punktów na mecz w sezonie 1968-69. Ponieważ sposób gry od tego czasu zmienił się, średnia punktów na mecz z roku na rok rośnie. Zawodnicy są coraz lepiej trenowani, a nowoczesna koszykówka przykłada większą uwagę atakowi niż obronie. Zdecydowano się zatem ustawić próg na 13 *pts_per_game*. Nie przewiduje się w przyszłości spadku wartości tej statystyki u MVP, a raczej jej wzrost,
- duża ilość punktów zdobywanych na mecz, wiąże się również ze średnią ilością prób rzutowych gracza w każdym meczu. Usunięto 40 graczy, którzy oddają mniej niż 10 prób rzutowych na mecz, wykorzystując statystykę *fga_per_game*.
- aby próby rzutowe przeradzały się w punkty, powinny być poparte wysoką skutecznością. Za pomocą statystyki *fg%_total*, usunięto pozostałych w zbiorze 5 zawodników, notujących skuteczność z gry poniżej 38% w całym sezonie. Wartość wynika z obniżenia średniej drużynowej skuteczności rzutów z gry, na sezon 2023-2024 (48%), o 10 punktów procentowych,
- tak jak już wcześniej wspomniano w rozdziale 1 sekcji 1.1, przy wyborze MVP, zwraca się uwagę również na wpływ zawodnika na resztę jego drużyny. Usunięto zatem 277

graczy którzy w danych sezonach rozdawali mniej niż 2 asysty na mecz, wykorzystując statystykę *ast_per_game*,

- zawodnik pokroju MVP również prezentować ponadprzeciętne umiejętności po obu stronach boiska. Pozostawiono w zbiorze zatem tylko koszykarzy, notujących powyżej 3 zbiórek na mecz, opierając się na statystyce *trb_per_game*. Poniżej tego progu znajdowało się 107 zawodników,
- ostatecznie, usunięto graczy będących na boisku mniej niż 30 minut na mecz. Standardowy czas meczu bez dogrywki, to 48 minut. Koszykarz, który może być kandydatem na *MVP*, powinien być wykorzystywany przez trenera przez znaczną większość meczu. Usunięto ostatnich 90 zawodników poniżej tej wartości progowej, statystyki *mp_per_game*.

Filtrowanie progowe pozwoliło na pozostawienie 1013 próbek zawodników w zbiorze treningowym, co pokazane jest na Rys. 2.8. Chciano jednak, żeby model mógł być końcowo skutecznie wykorzystywany do predykcji *MVP* z całego zbioru danych z kolejnych sezonów regularnych, bez wcześniejszego usuwania z nich rekordów. Zbiór testowy pozostawiono zatem w niezmienionej postaci 2013 rekordów, mając na uwadze zachwianie proporcji oraz możliwość zawyżenia wartości *RMSE* dla predykcji na samym zbiorze testowym.



Rys. 2.8. Ilość zawodników w zbiorze treningowym po filtracji progowej w zależności od wartości *MVP Votes Share*

Pozostała jeszcze kwestia silnej przynależności modelu do jednej cechy. Jak narazie, model nie miał żadnej możliwości wyboru parametrów najlepszych do predykcji, a korzystał z całej ich grupy dostępnej w zbiorze danych. Postanowiono zatem dostosować proces odszukiwania najlepszej grupy hiperparametrów, wprowadzając ilość wykorzystywanych cech, jako parametr zawarty w *param grid*. *Random Forest Regressor* nie oferuje jednak tego typu rozwiązania. Skorzystano zatem z metody selekcji cech *SelectKBest*, dostępnej w bibliotece *sckit-learn*. Pomaga ona wybrać określoną liczbę cech dających najlepsze wyniki, na podstawie zadanej funkcji oceny, używając parametru *score_func*. Jako funkcję oceny wybrano *f_regression*, obliczającą współczynniki statystyczne F, które służą do oceny zależności między cechą a zmienną predyktowaną. Test F porównuje wariancję międzygrupową (miarą dopasowania modelu) z wariancją wewnątrzgrupową (miarą błędu modelu). Współczynnik F dla każdej cechy jest obliczany według wzoru (2.4.1)

$$F = \frac{\frac{\text{SST}}{p}}{\frac{\text{SSE}}{n-p-1}} \tag{2.4.1}$$

Opis zmiennych:

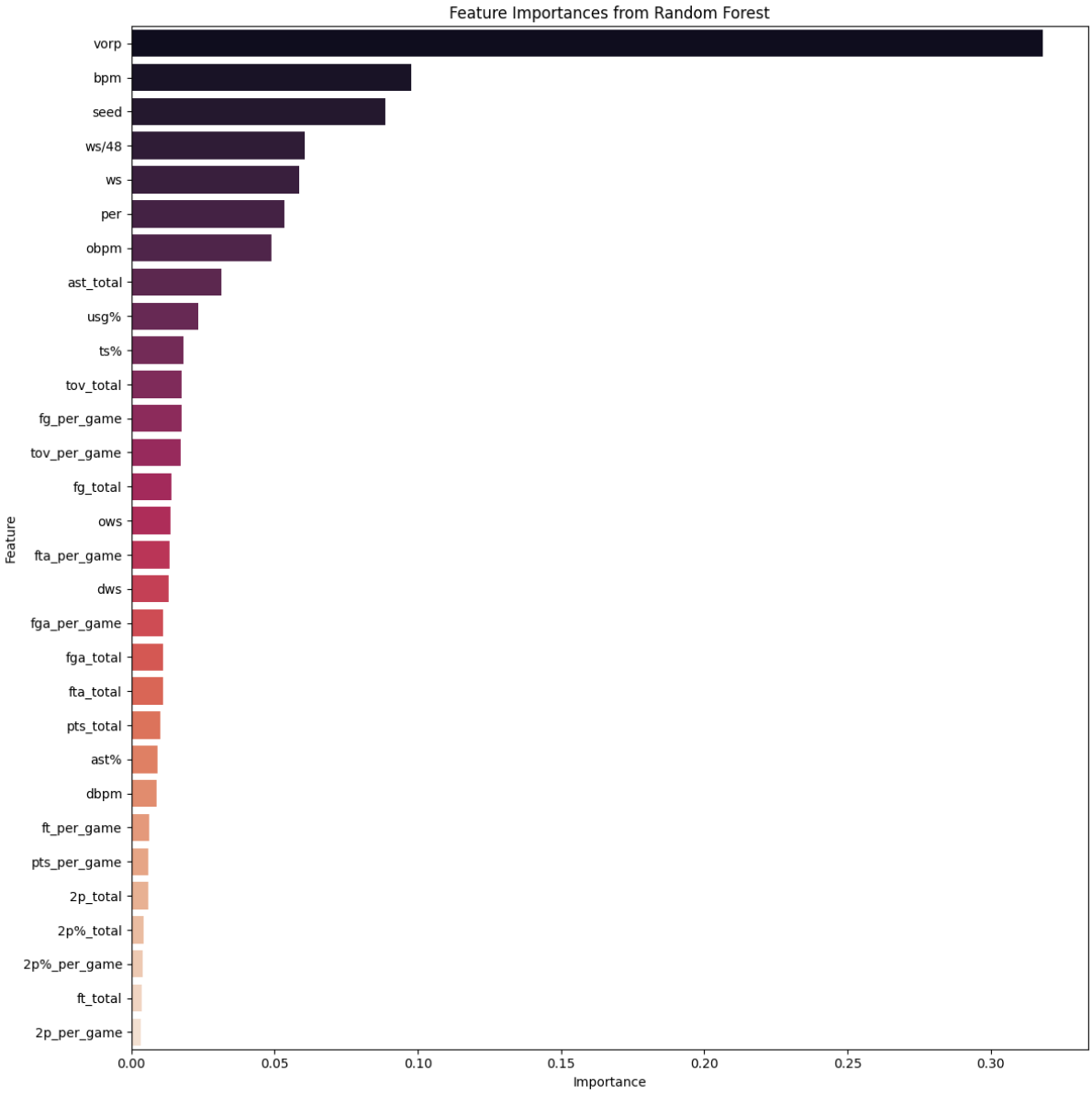
- *SST* - suma kwadratów całkowitych, czyli całkowita wariancja zmiennej prognozowanej. Oblicza się ją jako sumę kwadratów różnic między rzeczywistymi wartościami, a średnią zmiennej zależnej predyktowanej
- *SSE* - suma kwadratów reszt, czyli wariancja, która nie została wyjaśniona przez model. Jest to miara błędu, czyli różnicy między rzeczywistymi wartościami, a przewidywanymi przez model. Oblicza się ją jako sumę kwadratów różnic rzeczywistych wartości zmiennej prognozowanej, a przewidywanymi wartościami przez model
- p - liczba parametrów modelu (liczba cech)
- n - liczba próbek

Wartość F wskazuje, czy dana cecha niesie istotną informację dla modelu. Im wyższa wartość F, tym bardziej istotna jest cecha dla przewidywania. W kontekście funkcji *f_regression*, cechy o wyższej wartości F zostaną zatem uznane za bardziej istotne i wybrane do predykcji w określonej wcześniej liczbie.

Ponieważ wybór ostatecznej grupy cech i dobór ich liczby nie są intuicyjne, podjęto decyzję o dodaniu liczby kluczowych cech jako sztucznego hiperparametru w *param grid*. Skorzystano zatem z klasy Pipeline również z biblioteki *sckit-learn*, pozwalającej na sekwencyjne dodawanie i łączenie kolejnych etapów przetwarzania danych oraz modelowania. Następnie dzięki jej pomocy dodano nowy hiperparametr, o nazwie *feature_selection__k*. *K* jest liczbą najlepszych cech, wybraną przez funkcję *SelectKBest* umieszczoną w *Pipeline*. W *param grid*, umieszczono natomiast listę wartości ilości cech do sprawdzenia przez *GridSearchCV*, zajmujące przedział od 10, do 70 cech z krokiem 10. Dzięki temu rozwiązaniu udało się uwzględnić zarówno liczbę, jak i wybór najlepszych cech dla modelu, biorąc pod uwagę, że ich liczba oraz rodzaj mogą się zmieniać w zależności od doboru innych hiperparametrów, których zakres wartości został wcześniej określony. Po wprowadzeniu potrzebnych poprawek, uruchomiono algorytm *GridSearchCV* w celu znalezienia nowego najlepszego estymatora. Uczenie modeli we wszystkich możliwościach zajęło dużo więcej czasu, ze względu na dodanie nowych 7 wartości dla parametru *feature_selection__k*. Ostatecznie najlepsze wyniki okazał się dawać model o nastepującej grupie wartości hiperparametrów:

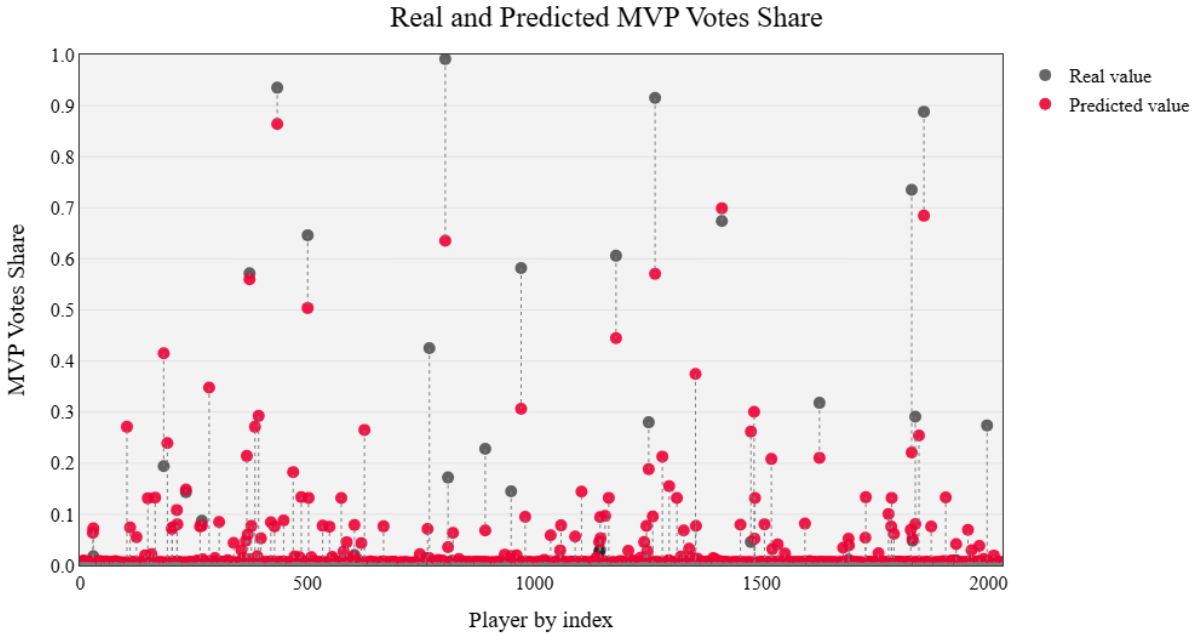
- *feature_selection__k* : 30
- *n_estimators* : 50
- *max_depth* : 19
- *max_samples* : 0.9
- *bootstrap* : True

Wartość *RMSE* dla zbioru treningowego wzrosła nieznacznie z 0.019 w przypadku pierwszego modelu, do 0.055 w przypadku obecnego. Wzrost nie jest jednak duży, a wynik zadowalający. Powodem pogorszenia się wartości funkcji celu, jest z pewnością usunięcie znacznej ilości próbek o *MVP Votes Share* = 0.0. Ponieważ stanowiły one grupę większościową, a co za tym idzie, prostszą do przewidzenia, model w zbiorze treningowym może sobie radzić trochę gorzej. Wartość *RMSE* dla zbioru testowego utrzymała się na poziomie podobnym do pierwszego modelu, przyjmując zadowalającą wartość 0.036. Algorytm *GridSearch* pozwolił także wreszcie na wybór określonej grupy najważniejszych cech, które zostały przedstawione na Rys 2.9. Widać wyraźną poprawę względem poprzedniego modelu. Nie dość, że usunięta została ponad połowa cech, które nie wnoszą ważnych informacji w kwestii predykcji, to dodatkowo obniżono częściowo dominację atrybutu *VORP* (z wartości *importance* wyższej niż 0.5, do około 0.3).

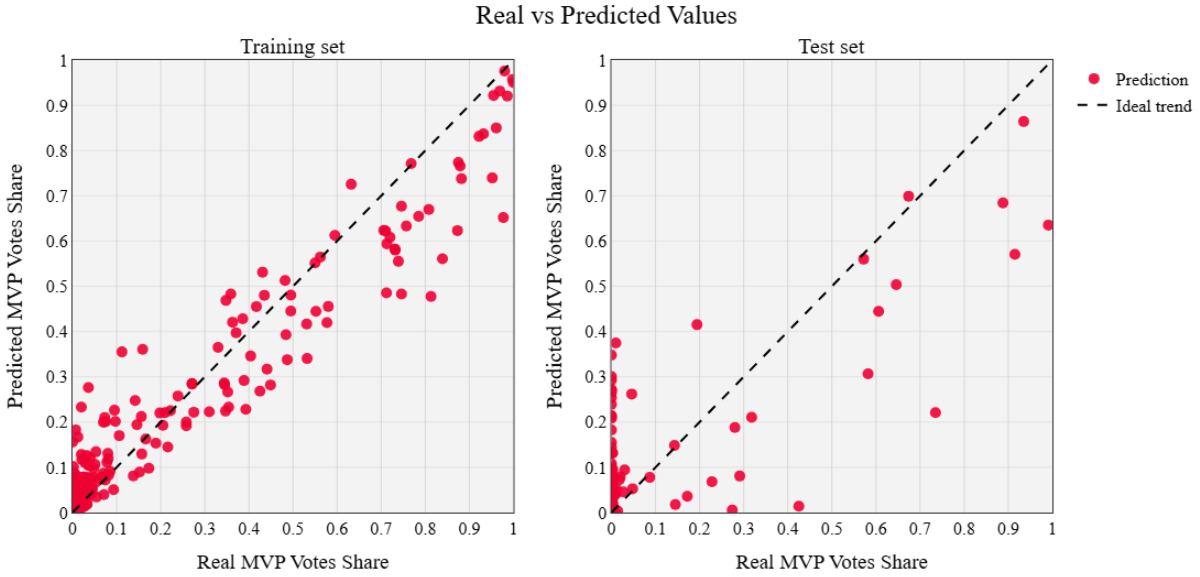


Rys. 2.9. Wykres wybranej grupy najważniejszych cech dla modelu 2

Pozostałe cechy również pozmieniały wartości swoje wagi, względem pierwszego modelu. Znacznie wzrosło np. znaczenie cechy *bpm* (z miejsca 13 na miejsce 2), oceniającej wpływ zawodnika na wynik drużyny podczas jego obecności na boisku. Ważne dla modelu cechy pozostały jednak blisko swoich poprzednich pozycji. Cecha *seed*, czyli miejsce drużyny w tabeli klasyfikacyjnej sezonu, pozostała jedną z najważniejszych dla modelu, znajdując się dalej w pierwszej trójce. Wykres 2.10 nie wykazał jednak znacznej zmiany w skuteczności predykcji. Model 2 dalej nieprecyzyjnie przewidywał wartości *MVP Votes Share*.

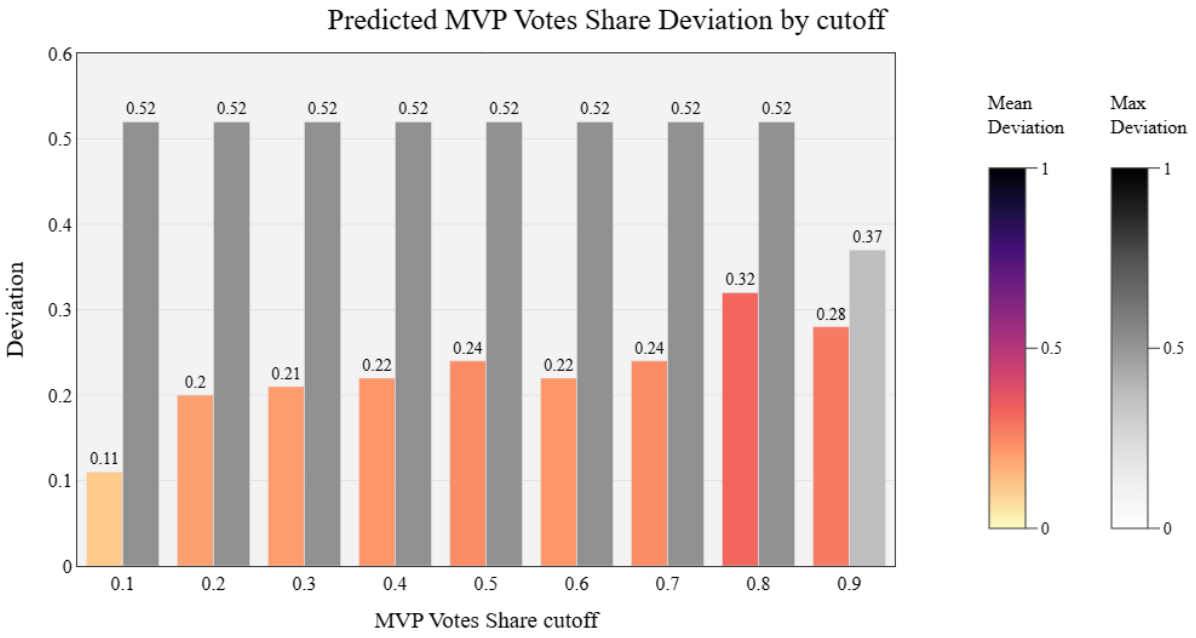


Rys. 2.10. Porównanie realnych i prognozowanych przez model 1 wartości MVP Votes Share na zbiorze testowym



Rys. 2.11. Porównanie realnych i prognozowanych przez model 2 wartości MVP Votes Share na zbiorach treningowym i testowym

Ponad to, na wykresach 2.10 oraz 2.11 widać, że pomimo znacznej poprawy precyzji modelu na zbiorze treningowym, nie poprawia się ona niestety na zbiorze testowym. Jest tutaj brany pod uwagę fakt, że duża część próbek ze zbioru testowego, ma wartość *MVP Votes Share* równą 0.0, w przeciwieńswie do zbioru trenigowego. Tłumaczy to więc zawyżanie ich wartości, widoczne w lewym dolnym rogu wykresu dotyczącego zbioru testowego. Niepokojąca jest jednak niska precyzja dla najważniejszej grupy graczy, posiadających wysoką wartość predyktora. Rozrzut wartości prognozowanych na zbiorze testowym nie zmniejszył się, co mogłoby sugerować *overfitting*, w połączeniu z obiecującym uogólnieniem na zbiorze testowym. Na wykresie 2.12 co ciekawe nie widać żadnej poprawy względem poprzedniego modelu, a nawet delikatne pogorszenie precyzji dla najwyższego przedziału wartości *MVP Votes Share*.



Rys. 2.12. Wykres średnich i maksymalnych dewiacji pomiędzy predykcjami modelu 2 a wartościami realnymi

Częściowa poprawa modelu względem poprzedniego, wywołała jeszcze chęć porównania modeli pierwszego i drugiego, pod kluczowym względem - predykcji najlepszych 5 graczy dla każdego sezonu, a w tym predykcji najlepszego z nich, czyli prognozy *MVP*. Napisano zatem kod pozwalający na przegląd i porównanie prawdziwych zestawień, do zestawień stworzonych na podstawie posortowania predyktowanych wartości *MVP Votes Share* w danym sezonie. Opracowano 4 zestawienia dla zbioru testowego oraz 19 dla zbioru treningowego, każde dotyczące osobnego sezonu regularnego, tak jak opisano to w sekcji 2.1. Na Rys. 2.13 pokazano strukturę takiego porównania, dla pojedynczego sezonu.

2023 Real		2023 Predicted	
player	mvp votes share	player	pred

Rys. 2.13. Struktura tabeli porównawczej rankingów

Wykonując powyższe czynności kolejno dla pierwszego i drugiego modelu, można było porównać ich skuteczność zadaniu prognozowania *MVP*. Skompletowane zestawienia obydwu modeli można zobaczyć na Rys. 2.14 oraz 2.15. Każda poprawna predykcja zawodnika MVP dla danego sezonu została oznaczona kolorem zielonym. Niepoprawny wybór najbardziej wartościowego zawodnika oznaczono natomiast kolorem czerwonym, a obywie obserwacje można było porównać do realnego *MVP* zaznaczonego kolorem szarym. Już na pierwszy rzut oka było widać, że pomimo wysokich dewiacji wartości predyktowanych od realnych, stosunkowo wysokich błędów maksymalnych w określonych przedziałach *MVP Votes Share* oraz wyraźnej dominacji jednej lub kilku atrybutów, obydwa modele całkiem dobrze radziły sobie z predykcją *MVP*. Porównano zatem model pierwszy i drugi, za pomocą wcześniej niestosowanych miar - ilości poprawnie wybranych *MVP* oraz sumarycznej ilości koszykarzy poprawnie przewidzianych do znajdowania się w pierwszej piątce rankingu.

TRAIN DATASET

2023 Real		2023 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.935	nikola jokić	0.883
shai gilgeous-alexander	0.646	giannis antetokounmpo	0.534
luka dončić	0.572	luka dončić	0.507
giannis antetokounmpo	0.194	shai gilgeous-alexander	0.456
jalen brunson	0.143	joel embiid	0.392

2022 Real		2022 Predicted	
player	mvp votes share	player	pred
joel embiid	0.915	nikola jokić	0.648
nikola jokić	0.674	joel embiid	0.455
giannis antetokounmpo	0.606	giannis antetokounmpo	0.372
jayson tatum	0.28	luka dončić	0.351
shai gilgeous-alexander	0.046	tyler dorsey	0.303

2011 Real		2011 Predicted	
player	mvp votes share	player	pred
lebron james	0.888	lebron james	0.56
kevin durant	0.735	chris paul	0.211
chris paul	0.318	kyrylo fesenko	0.192
kobe bryant	0.291	rajon rondo	0.179
tony parker	0.274	kevin durant	0.168

2003 Real		2003 Predicted	
player	mvp votes share	player	pred
kevin garnett	0.991	kevin garnett	0.542
tim duncan	0.582	tim duncan	0.332
jermaine o'neal	0.425	tracy mcgrady	0.167
peja stojaković	0.228	bruno šundov	0.119
kobe bryant	0.172	shaquille o'neal	0.117

TEST DATASET

2021 Real		2021 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.875	nikola jokić	0.684
joel embiid	0.706	giannis antetokounmpo	0.572
giannis antetokounmpo	0.595	joel embiid	0.462
devin booker	0.216	luka dončić	0.259
luka dončić	0.146	lebron james	0.163

2020 Real		2020 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.961	nikola jokić	0.677
joel embiid	0.58	joel embiid	0.349
stephen curry	0.449	stephen curry	0.312
giannis antetokounmpo	0.345	giannis antetokounmpo	0.287
chris paul	0.138	damian lillard	0.142

2019 Real		2019 Predicted	
player	mvp votes share	player	pred
giannis antetokounmpo	0.952	giannis antetokounmpo	0.618
lebron james	0.746	james harden	0.455
james harden	0.363	lebron james	0.357
luka dončić	0.198	luka dončić	0.241
kawhi leonard	0.166	damian lillard	0.21

2018 Real		2018 Predicted	
player	mvp votes share	player	pred
giannis antetokounmpo	0.932	james harden	0.809
james harden	0.768	giannis antetokounmpo	0.687
paul george	0.352	paul george	0.269
nikola jokić	0.21	nikola jokić	0.229
stephen curry	0.173	stephen curry	0.134

2017 Real		2017 Predicted	
player	mvp votes share	player	pred
james harden	0.955	james harden	0.756
lebron james	0.731	lebron james	0.576
anthony davis	0.441	anthony davis	0.266
damian lillard	0.205	russell westbrook	0.202
russell westbrook	0.07	damian lillard	0.19

2016 Real		2016 Predicted	
player	mvp votes share	player	pred
russell westbrook	0.879	james harden	0.642
james harden	0.746	russell westbrook	0.606
kawhi leonard	0.495	kawhi leonard	0.452
lebron james	0.33	lebron james	0.392
isaiah thomas	0.08	kevin durant	0.252

2015 Real		2015 Predicted	
player	mvp votes share	player	pred
stephen curry	1.0	stephen curry	0.927
kawhi leonard	0.484	lebron james	0.426
lebron james	0.482	russell westbrook	0.374
russell westbrook	0.371	kawhi leonard	0.352
kevin durant	0.112	kevin durant	0.295

2014 Real		2014 Predicted	
player	mvp votes share	player	pred
stephen curry	0.922	stephen curry	0.708
james harden	0.72	james harden	0.601
lebron james	0.425	anthony davis	0.308
russell westbrook	0.271	russell westbrook	0.289
anthony davis	0.156	lebron james	0.284

2013 Real		2013 Predicted	
player	mvp votes share	player	pred
kevin durant	0.986	kevin durant	0.905
lebron james	0.713	lebron james	0.609
blake griffin	0.347	kevin love	0.242
joakim noah	0.258	blake griffin	0.224
james harden	0.068	stephen curry	0.183

2012 Real		2012 Predicted	
player	mvp votes share	player	pred
lebron james	0.998	lebron james	0.792
kevin durant	0.632	kevin durant	0.645
carmelo anthony	0.393	chris paul	0.329
chris paul	0.239	carmelo anthony	0.163
kobe bryant	0.152	james harden	0.148

2010 Real		2010 Predicted	
player	mvp votes share	player	pred
derrick rose	0.977	lebron james	0.505
dwight howard	0.531	derrick rose	0.465
lebron james	0.431	dwight howard	0.342
kobe bryant	0.354	dwyane wade	0.206
kevin durant	0.157	kobe bryant	0.156

2009 Real		2009 Predicted	
player	mvp votes share	player	pred
lebron james	0.98	lebron james	0.914
kevin durant	0.495	kevin durant	0.515
kobe bryant	0.487	dwight howard	0.31
dwight howard	0.389	dwyane wade	0.307
dwyane wade	0.097	kobe bryant	0.274

2008 Real		2008 Predicted	
player	mvp votes share	player	pred
lebron james	0.969	lebron james	0.931
kobe bryant	0.577	dwyane wade	0.593
dwyane wade	0.562	chris paul	0.531
dwight howard	0.271	kobe bryant	0.449
chris paul	0.159	dwight howard	0.279

2007 Real		2007 Predicted	
player	mvp votes share	player	pred
kobe bryant	0.873	chris paul	0.549
chris paul	0.71	lebron james	0.456
kevin garnett	0.532	kobe bryant	0.431
lebron james	0.348	kevin garnett	0.264
dwight howard	0.048	amar'e stoudemire	0.224

2006 Real		2006 Predicted	
player	mvp votes share	player	pred
dirk nowitzki	0.882	dirk nowitzki	0.673
steve nash	0.785	steve nash	0.644
kobe bryant	0.404	kobe bryant	0.378
tim duncan	0.222	lebron james	0.26
lebron james	0.142	tim duncan	0.253

2005 Real		2005 Predicted	
player	mvp votes share	player	pred
steve nash	0.739	lebron james	0.51
lebron james	0.55	dirk nowitzki	0.447
dirk nowitzki	0.435	kobe bryant	0.419
kobe bryant	0.386	steve nash	0.366
chauncey billups	0.344	dwyane wade	0.343

2004 Real		2004 Predicted	
player	mvp votes share	player	pred
steve nash	0.839	lebron james	0.326
shaquille o'neal	0.813	shaquille o'neal	0.302
dirk nowitzki	0.275	steve nash	0.289
tim duncan	0.258	dirk nowitzki	0.286
allen iverson	0.189	kevin garnett	0.271

2002 Real		2002 Predicted	
player	mvp votes share	player	pred
tim duncan	0.808	tracy mcgrady	0.494
kevin garnett	0.732	tim duncan	0.44
kobe bryant	0.417	kobe bryant	0.435
tracy mcgrady	0.359	kevin garnett	0.412
shaquille o'neal	0.106	dirk nowitzki	0.305

2001 Real		2001 Predicted	
player	mvp votes share	player	pred
tim duncan	0.757	tim duncan	0.681
jason kidd	0.712	shaquille o'neal	0.45
shaquille o'neal	0.552	jason kidd	0.421
tracy mcgrady	0.31	paul pierce	0.252
kobe bryant	0.078	tracy mcgrady	0.21

Rys. 2.14. Wyniki predykcji modelu 1

TRAIN DATASET

2023 Real		2023 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.935	nikola jokić	0.883
shai gilgeous-alexander	0.646	shai gilgeous-alexander	0.531
luka dončić	0.572	luka dončić	0.483
giannis antetokounmpo	0.194	giannis antetokounmpo	0.458
jalen brunson	0.143	joel embiid	0.385

2022 Real		2022 Predicted	
player	mvp votes share	player	pred
joel embiid	0.915	nikola jokić	0.651
nikola jokić	0.674	joel embiid	0.607
giannis antetokounmpo	0.606	giannis antetokounmpo	0.446
jayson tatum	0.28	luka dončić	0.433
shai gilgeous-alexander	0.046	stanley umude	0.311

2011 Real		2011 Predicted	
player	mvp votes share	player	pred
lebron james	0.888	lebron james	0.522
kevin durant	0.735	kevin durant	0.238
chris paul	0.318	kyrylo fesenko	0.237
kobe bryant	0.291	chris paul	0.187
tony parker	0.274	derrick rose	0.156

2003 Real		2003 Predicted	
player	mvp votes share	player	pred
kevin garnett	0.991	kevin garnett	0.645
tim duncan	0.582	tim duncan	0.389
jermaine o'neal	0.425	bruno šundov	0.256
peja stojaković	0.228	tracy mcgrady	0.104
kobe bryant	0.172	andrei kirilenko	0.099

TEST DATASET

2021 Real		2021 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.875	nikola jokić	0.704
joel embiid	0.706	giannis antetokounmpo	0.625
giannis antetokounmpo	0.595	joel embiid	0.622
devin booker	0.216	luka dončić	0.183
luka dončić	0.146	devin booker	0.12

2020 Real		2020 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.961	nikola jokić	0.85
joel embiid	0.58	joel embiid	0.426
stephen curry	0.449	stephen curry	0.38
giannis antetokounmpo	0.345	giannis antetokounmpo	0.319
chris paul	0.138	chris paul	0.074

2019 Real		2019 Predicted	
player	mvp votes share	player	pred
giannis antetokounmpo	0.952	giannis antetokounmpo	0.618
lebron james	0.746	james harden	0.455
james harden	0.363	lebron james	0.357
luka dončić	0.198	luka dončić	0.241
kawhi leonard	0.166	damian lillard	0.21

2018 Real		2018 Predicted	
player	mvp votes share	player	pred
giannis antetokounmpo	0.932	james harden	0.76
james harden	0.768	giannis antetokounmpo	0.723
paul george	0.352	nikola jokić	0.259
nikola jokić	0.21	paul george	0.24
stephen curry	0.173	stephen curry	0.133

2017 Real		2017 Predicted	
player	mvp votes share	player	pred
james harden	0.955	james harden	0.848
lebron james	0.731	lebron james	0.642
anthony davis	0.441	anthony davis	0.251
damian lillard	0.205	russell westbrook	0.169
russell westbrook	0.07	damian lillard	0.161

2016 Real		2016 Predicted	
player	mvp votes share	player	pred
russell westbrook	0.879	james harden	0.695
james harden	0.746	russell westbrook	0.67
kawhi leonard	0.495	kawhi leonard	0.456
lebron james	0.33	lebron james	0.301
isaiah thomas	0.08	kevin durant	0.168

2015 Real		2015 Predicted	
player	mvp votes share	player	pred
stephen curry	1.0	stephen curry	0.927
kawhi leonard	0.484	lebron james	0.426
lebron james	0.482	russell westbrook	0.374
russell westbrook	0.371	kawhi leonard	0.352
kevin durant	0.112	kevin durant	0.295

2014 Real		2014 Predicted	
player	mvp votes share	player	pred
stephen curry	0.922	stephen curry	0.865
james harden	0.72	james harden	0.645
lebron james	0.425	russell westbrook	0.266
russell westbrook	0.271	lebron james	0.262
anthony davis	0.156	anthony davis	0.239

2013 Real		2013 Predicted	
player	mvp votes share	player	pred
kevin durant	0.986	kevin durant	0.905
lebron james	0.713	lebron james	0.609
blake griffin	0.347	kevin love	0.242
joakim noah	0.258	blake griffin	0.224
james harden	0.068	stephen curry	0.183

2012 Real		2012 Predicted	
player	mvp votes share	player	pred
lebron james	0.998	lebron james	0.911
kevin durant	0.632	kevin durant	0.686
carmelo anthony	0.393	carmelo anthony	0.287
chris paul	0.239	chris paul	0.257
kobe bryant	0.152	kobe bryant	0.09

2010 Real		2010 Predicted	
player	mvp votes share	player	pred
derrick rose	0.977	lebron james	0.678
dwight howard	0.531	derrick rose	0.529
lebron james	0.431	dwight howard	0.424
kobe bryant	0.354	kobe bryant	0.257
kevin durant	0.157	dwyane wade	0.155

2009 Real		2009 Predicted	
player	mvp votes share	player	pred
lebron james	0.98	lebron james	0.968
kevin durant	0.495	kevin durant	0.374
kobe bryant	0.487	kobe bryant	0.328
dwight howard	0.389	dwight howard	0.283
dwyane wade	0.097	steve nash	0.184

2008 Real		2008 Predicted	
player	mvp votes share	player	pred
lebron james	0.969	lebron james	0.919
kobe bryant	0.577	dwyane wade	0.504
dwyane wade	0.562	kobe bryant	0.428
dwight howard	0.271	chris paul	0.338
chris paul	0.159	dwight howard	0.298

2007 Real		2007 Predicted	
player	mvp votes share	player	pred
kobe bryant	0.873	chris paul	0.608
chris paul	0.71	kobe bryant	0.603
kevin garnett	0.532	kevin garnett	0.41
lebron james	0.348	lebron james	0.376
dwight howard	0.048	dwight howard	0.18

2006 Real		2006 Predicted	
player	mvp votes share	player	pred
dirk nowitzki	0.882	dirk nowitzki	0.698
steve nash	0.785	steve nash	0.683
kobe bryant	0.404	kobe bryant	0.38
tim duncan	0.222	lebron james	0.257
lebron james	0.142	tim duncan	0.24

2005 Real		2005 Predicted	
player	mvp votes share	player	pred
steve nash	0.739	lebron james	0.524
lebron james	0.55	steve nash	0.514
dirk nowitzki	0.435	dirk nowitzki	0.468
kobe bryant	0.386	kobe bryant	0.43
chauncey billups	0.344	chauncey billups	0.29

2004 Real		2004 Predicted	
player	mvp votes share	player	pred
steve nash	0.839	steve nash	0.637
shaquille o'neal	0.813	shaquille o'neal	0.519
dirk nowitzki	0.275	tim duncan	0.205
tim duncan	0.258	dirk nowitzki	0.201
allen iverson	0.189	allen iverson	0.18

2002 Real		2002 Predicted	
player	mvp votes share	player	pred
tim duncan	0.808	tim duncan	0.657
kevin garnett	0.732	kevin garnett	0.596
kobe bryant	0.417	kobe bryant	0.474
tracy mcgrady	0.359	tracy mcgrady	0.452
shaquille o'neal	0.106	dirk nowitzki	0.251

2001 Real		2001 Predicted	
player	mvp votes share	player	pred
tim duncan	0.757	tim duncan	0.679
jason kidd	0.712	shaquille o'neal	0.511
shaquille o'neal	0.552	jason kidd	0.484
tracy mcgrady	0.31	tracy mcgrady	0.24
kobe bryant	0.078	kobe bryant	0.1

Rys. 2.15. Wyniki predykcji modelu 2

Podsumowując zestawienia 2.14 i 2.15 otrzymano wyniki zebrane w tabelach 2.3 i 2.4:

	Poprawnych predykcji	**Wszystkich predykcji**	**Skuteczność**
MVP	15	23	62.5%
Top 5 zawodników	91	115	79.1%

Tabela 2.3. Model 1 - tabela skuteczności predykcji

	Poprawnych predykcji	**Wszystkich predykcji**	**Skuteczność**
MVP	17	23	73.9%
Top 5 zawodników	99	115	86.0%

Tabela 2.4. Model 2 - tabela skuteczności predykcji

Model 2 wyraźnie poprawił efektywność obydwóch aspektów predykcji. Skuteczność ważniejszego z nich, czyli poprawnej prognozy *MVP*, wzrosła o ponad 10 punktów procentowych, poprawiając predykcję w dwóch sezonach (2002-03 oraz 2004-05). Rezultaty widoczne są na Rys. 2.16 2.17.

MVP Ranking in 2002-2003 season					
Real Ranking		**Model 1 Pred Ranking**		**Model 2 Pred Ranking**	
player	**mvp votes share**	**player**	**pred**	**player**	**pred**
tim duncan	0.808	tracy mcgrady	0.494	tim duncan	0.657
kevin garnett	0.732	tim duncan	0.44	kevin garnett	0.596
kobe bryant	0.417	kobe bryant	0.435	kobe bryant	0.474
tracy mcgrady	0.359	kevin garnett	0.412	tracy mcgrady	0.452
shaquille o'neal	0.106	dirk nowitzki	0.305	dirk nowitzki	0.251

Rys. 2.16. Porównanie predykcji rankingu plebiscytu MVP w sezonie 2002-03

MVP Ranking in 2004-2005 season					
Real Ranking		**Model 1 Pred Ranking**		**Model 2 Pred Ranking**	
player	**mvp votes share**	**player**	**pred**	**player**	**pred**
steve nash	0.839	lebron james	0.326	steve nash	0.637
shaquille o'neal	0.813	shaquille o'neal	0.302	shaquille o'neal	0.519
dirk nowitzki	0.275	steve nash	0.289	tim duncan	0.205
tim duncan	0.258	dirk nowitzki	0.286	dirk nowitzki	0.201
allen iverson	0.189	kevin garnett	0.271	allen iverson	0.18

Rys. 2.17. Porównanie predykcji rankingu plebiscytu MVP w sezonie 2004-05

Zauważono również poprawę precyzji umiejscowienia zawodników na kolejnych miejscach rankingu, względem pierwszego modelu. W sezonie 2011-12 poprawnie model 1 pozwolił na poprawną predykcję *MVP*, natomiast zawodnik zajmujący realnie, bardzo bliskie, drugie miejsce w rankingu - Kevin Durant - został przypisany do miejsca piątego. Model 2 poprawił prognozę i umiejscowił Kevin'a Duranta na poprawnej pozycji (Rys 2.18). Należy jednak zaznaczyć że wartość jego *MVP Votes Share* dalej była daleka od zbliżenia się do wygrania plebiscytu *MVP*, w przeciwieństwie do realnych danych.

MVP Ranking in 2011-2012 season					
Real Ranking		Model 1 Pred Ranking		Model 2 Pred Ranking	
player	mvp votes share	player	pred	player	pred
lebron james	0.888	lebron james	0.56	lebron james	0.522
kevin durant	0.735	chris paul	0.211	kevin durant	0.238
chris paul	0.318	kyrylo fesenko	0.192	kyrylo fesenko	0.237
kobe bryant	0.291	rajon rondo	0.179	chris paul	0.187
tony parker	0.274	kevin durant	0.168	derrick rose	0.156

Rys. 2.18. Poprawa predykcji drugiej pozycji w rankingu *MVP* w sezonie 2011-12

Dopiero korzystając z tego typu porównania wyraźnie widać było, że wprowadzone adaptacje pozytywnie wpłynęły na predykcję. Model 2 dalej nie prezentował jednak wystarczająco zadowalających wyników oraz wykazywał potencjalnie niepokojące, niezrozumiałe zachowania. Przykładem mógł być ranking sezonu 2022-23, w którym model 2 na piątym miejscu prognozował Stanley'a Umude, nie znajdującego się w ogóle w realnym rankingu z tego sezonu. Ponad to, koszykarz ten rzucał w tym sezonie średnio 2 punkty na mecz, grając dla jednej z najgorszych drużyn w *NBA*, która ostatecznie znalazła się na końcu tabeli. Model 2 przyznał temu zawodnikowi wysoką wartość *MVP Votes Share* = 0.311, co biorąc pod uwagę prezencję jego drużyny, oraz słabe statystyki osobiste w ogóle nie powinno mieć miejsca (Rys 2.19).

2022 Real		2022 Predicted	
player	mvp votes share	player	pred
joel embiid	0.915	nikola jokić	0.651
nikola jokić	0.674	joel embiid	0.607
giannis antetokounmpo	0.606	giannis antetokounmpo	0.446
jayson tatum	0.28	luka dončić	0.433
shai gilgeous-alexander	0.046	stanley umude	0.311

Rys. 2.19. Niepokojąca predykcja piątego zawodnika rankingu *MVP* w sezonie 2022-23

Na zestawieniu 2.15, szczególnie w części odnoszącej się do zbioru testowego, zauważalne są przypadki przewidywanych rankingów, w których pojawiają się zawodnicy nieobecni w rzeczywistym rankingu. Niezbalansowanie zbioru doprowadziło do konieczności zastosowania filtracji i odrzucenia dużej ilości próbek o niskim współczynniku *MVP Votes Share*, przez to model dopasował się zbytnio do zawodników o jego nieco wyższych wartościach. Pomimo całkiem zadowalających skuteczności procentowych, tego typu zachowania mogłyby spowodować niepoprawne predykcje w przyszłych sezonach, czego wolano uniknąć. Podjęto zatem działania mające na celu znalezienie przyczyny problemów związanych z dyskusyjną precyzją modelu regresyjnego. Zidentyfikowano w końcu, że obydwu modelom został założony niepoprawny sposób oceny skuteczności. Przewidywanie konkretnych wartości współczynnika *MVP Votes Share* było jak najbardziej w porządku, natomiast odzwierciedlająca skuteczność modelu funkcja RMSE, nie mierzyła tak na prawdę tego, czego od modelu oczekiwano. Precyzyjna wartość predyktora nie była w zasadzie do niczego potrzebna, a niezbalansowanie zbioru wpływało na niedokładność jej predykcji. Celem modelu nie powinna być prognoza dokładnej wartości wskaźnika *MVP Votes Share* dla każdego z zawodników. Model powinien natomiast dążyć do utworzenia jak najbardziej poprawnego rankingu *MVP*, a najlepszy z należących do niego koszykarzy powinien zostać wybrany najbardziej wartościowym graczem sezonu.

2.5. Ostateczny model

Ponieważ adaptacje wprowadzone podczas uczenia modelu 2 wyraźnie poprawiły jego ogólną precyzję, postanowiono pozostawić je także do uczenia modelu następnego. Poza filtracją progową oraz dodaniem *feature importance* jako hiperparametru, miała się zmienić jedynie funkcja celu. Znaleziono zatem odpowiednią i dopasowaną dokładnie do problemu funkcję, posiłkując się literaturą naukową. Biorąc pod uwagę, że model miał za zadanie ostatecznie generować ranking najlepszych pięciu graczy, wyróżniając przy tym najlepszego z nich jako *MVP*, zdecydowano, że jego dokładność powinna być oceniana dosłownie przez porównanie przewidzianego rankingu z rankingiem oryginalnym. Konkretna metoda porównywania list rankingowych została opisana w artykule „A Similarity Measure for Indefinite Rankings". Autorzy William Wevver, Alistair Moffat oraz Justin Zobel z Uniwersytetu w Melbourne, prezentowali w swoim artykule nową miarę podobieństwa rankingów [11]. Praca dotyczy zastosowania tej metody do porównywania niekompletnych list rankingowych, przykładając jednocześnie uwagę do pozycji na ich szczycie.

Ranking prognozowany przez modele opisane w tym projekcie dyplomowym nigdy nie będzie niekompletny, natomiast sam sposób porównania tych list, wydał się odpowiednią podstawą do zbudowania własnej funkcji celu. Metoda nosi nazwę *RBO* (*ang. Ranked Biased Overlap*), a wzór 2.5.1 opisuje jej działanie. Ze względu na długą nazwę metody, w dalszej części pracy posłużono się jej angielskim skrótem.

$$RBO(S,T,p) = (1-p)\sum_{d=1}^{\infty} p^{d-1} \cdot A(d) \tag{2.5.1}$$

Opis zmiennych:

- S, T - dwa rankingi, które są porównywane. S reprezentuje ranking przewidywany, a T ranking rzeczywisty (referencyjny),
- p - parametr stronniczości ($0 \leq p < 1$), który kontroluje wpływ pozycji w rankingu na wynik porównania. Im większe p, tym głębsze porównanie list rankingowych i mniejszy nacisk na zgodność w górnych partiach list,
- d - pozycja w rankingu, liczona od 1 do n, gdzie n jest długością rankingu,
- $A(d)$ - poziom zgodności elementów do pozycji d, między rankingami S i T, obliczany jako liczba wspólnych elementów w pierwszych d pozycjach obu rankingów, podzielona przez d.

RBO opiera się na założeniu, że wyższe pozycje na listach rankingowych są ważniejsze od niższych, co idealnie pasowało do problemu rozwiązywanego w tym projekcie. Teoretycznie za pomocą tej miary, można porównywać listy o dowolnej długości, takie jak np. wyniki wyszukiwania w przeglądarce. Rozkładając wzór (2.5.1) na czynniki pierwsze, działanie *RBO* można opisać w kilku krokach:

1. Zdefiniowanie parametru p - nazywany w artykule wytrwałością, określa jak szybko maleje waga niższych pozycji rankingu, co wynika wprost ze wzoru. Im mniejsze p, tym większa waga przypisywana jest zatem wyższym pozycjom.
2. Obliczanie zgodności list $A(d)$ na każdej możliwej głębokości - wartość $A(d)$ maleje z każdym kolejnym przejściem na większy poziom głębokości rankingu. Przykład: jeśli na poziomie $d = 3$ trzy pierwsze elementy obu list to kolejno *abc* oraz *adc*, to zgodność na tej głębokości wynosi $\frac{2}{3}$,

3. Ważenie zgodności - *RBO* przypisuje wagę do obliczonych wcześniej zgodności na każdej głębokości. Waga ta jest obliczana na podstawie parametru p i maleje geometrycznie wraz ze wzrostem głębokości. Waga dla głębokości d to: $(1-p)\cdot p^{d-1}$,
4. Sumowanie ważonych zgodności - *RBO* sumuje je na wszystkich głębokościach, aby uzyskać ostateczną ocenę podobieństwa list rankingowych. Sumowanie odbywa się teoretycznie do nieskończoności, natomiast w praktyce obliczenia wykonuje się do pewnej skończonej głębokości d

Wartość RBO mieści się w przedziale od 0 do 1, gdzie 0 oznacza całkowity brak podobieństwa, a 1 oznacza identyczność list [11]. W oparciu o opisaną powyżej miarę, zaprojektowano funkcję celu, dopasowaną do zadanego w pracy problemu. Stworzony *nba_rank_scorer* dla każdego sezonu generował najpierw prawdziwy ranking pięciu najlepszych zawodników pod względem wskaźnika *MVP Votes Share*, a następnie prognozowany ranking o tej samej długości, sortując zawodników malejąco według wartości predyktora. Funkcja celu porównywała oba rankingi, zawierające faktycznych oraz prognozowanych najlepszych pięciu koszykarzy z danego sezonu, obliczając następnie wartość RBO. Ostatecznym wynikiem działania funkcji był uśredniony wynik RBO, uzyskany dla wszystkich sezonów ze zbioru treningowego. Ze względu na to, że do uczenia wykorzystywano model *Random Forest*, wartość współczynnika p nie wpływała w żaden sposób na adaptację jego predykcji, a jedynie na ocene ostatecznych wyników i ewentualnie wybór innego estymatora. *Random Forest* nie ma bowiem wbudowanego mechanizmu typu propagacji wstecznej, co wyjaśniono szerzej w rozdziale 1 sekcji 1.3. Postanowiono zatem przetestować model na wartościach p równych kolejno 0.0, 0.3, 0.5 oraz 0.9, aby wybrać tę, która najlepiej odwzorowuje jakość prognzowanego rankingu, zapewniając najbardziej adekwatną do niego wartość współczynnika *RBO*. Wartość 0.0 np. bardzo surowo ocenia skuteczność modelu, nie zwracając przy okazji uwagi na żadne miejsca rankingu poza pierwszym - ocena jest zero-jedynkowa, *RBO*=0.0 w przypadku niepoprawnej predykcji pierwszej pozycji na liście lub *RBO*=1.0 w przypadku poprawnej. Po przeprowadzonych testach, wybrano wartość p=0.3, ponieważ nieco łagodniej oceniała model podczas niepoprawnych predykcji wysokich miejsc, dalej jednak przywiązując im znacznie większą wagę niż pozycjom niskim.

Uruchomiono zatem algorytm *GridSearch*, za pomocą którego wybrano najskuteczniejszy estymator o następujących hiperparametrach:

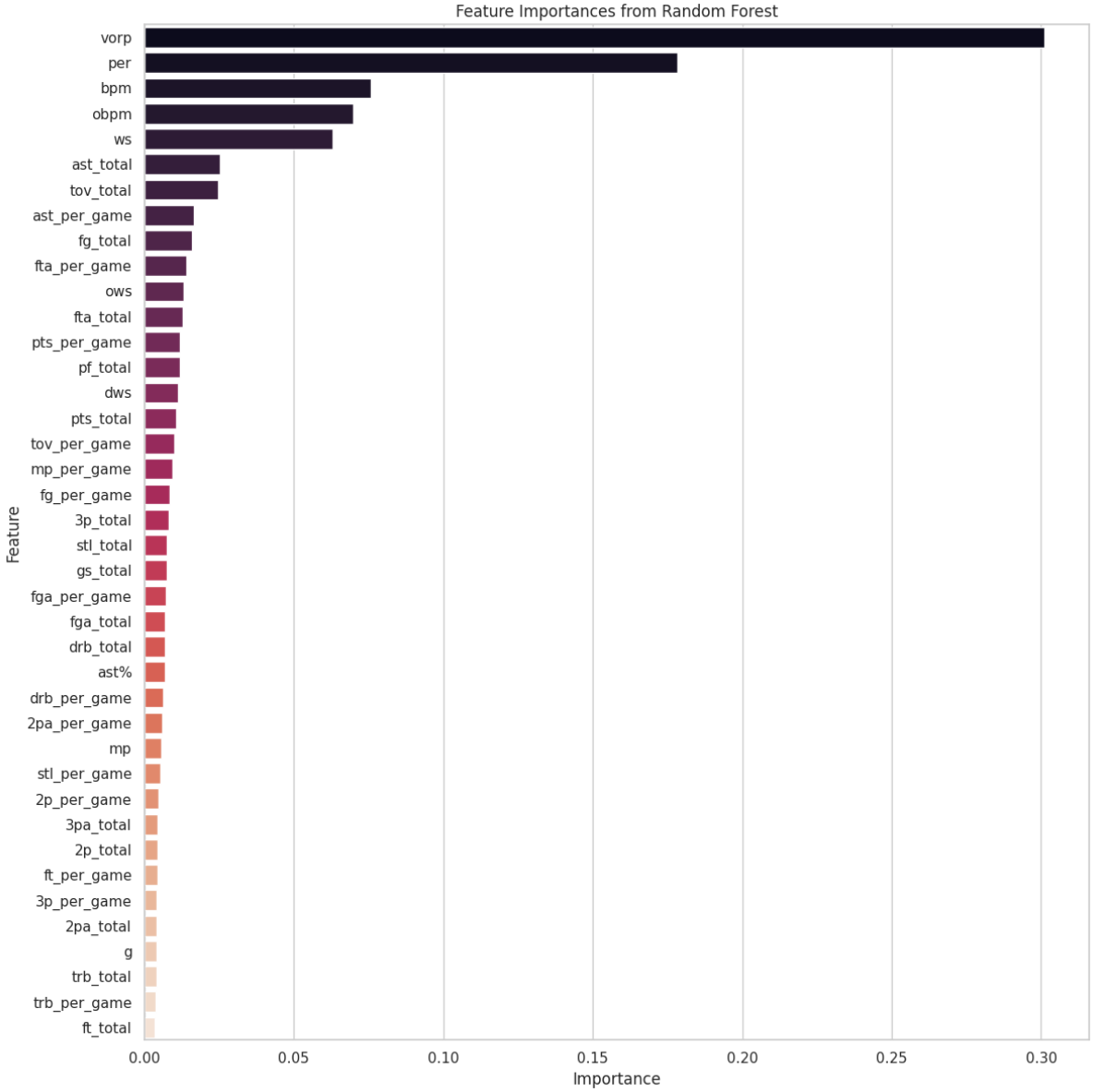
- *feature_selection__k* : 30
- *n_estimators* : 25
- *max_depth* : 7
- *max_samples* : 0.7
- *bootstrap* : True

Już dzięki samym wybranym ich wartościom, można było spodziewać się poprawy skuteczności nowego modelu. Potrzebował on o połowę mniej drzew decyzyjnych, a ich głębokość pomniejszyła się z 19 na 7. Świadczyło to o znacznie lepszym uogólnieniu problemu przez model, co powinno mieć przełożenie na precyzyjniejsze prognozy na zbiorze testowym. Ze względu na drastyczną zmianę podejścia przy wyborze funkcji kosztu, wcześniej stosowane wykresy nie mogły służyć już do weryfikacji jakości modelu. Skuteczność predykcji nie była w końcu mierzona poprzez porównanie konkretnych wartości *MVP Votes Share* z prognozowanymi, a przez wartość stopnia podobieństwa rankingów - *RBO*. Wykresy dewiacji i predykcji odrzucono zatem z etapu weryfikacji, zostawiając jedynie wykres *feature importance*. Postanowiono zweryfikować jakość modelu poprzez porównanie jego uśrednionej skuteczności do poprzednich, tworząc zestawienie predykcji identyczne w strukturze do obecnych na Rys. 2.14 i Rys. 2.15. Miało to pomóc ostatecznie podsumować poprawę jakości modelu używającego *RBO*, względem poprzednich.

2.6. Wyniki ostatecznego modelu

Model 3 dalej wykazywał silne przywiązanie do cechy *vorp*, zwiększając jednak między innymi wagę cech *per*, *bpm* oraz *obpm*. Mimo obrania lepszego, względem poprzedniego modelu, kierunku w kwestii doboru cech, ciekawym zachowaniem było całkowite odrzucenie atrybutu *seed* z grupy wykorzystywanych do predykcji. Atrybut ten informował przecież o ostatecznym miejscu w rankingu sezonu regularnego, zajętym przez drużynę zawodnika. Początkowo, wartość tej cechy wydawała się być kluczowa dla poprawnej predykcji, jednak podczas uczenia okazała się ona widocznie nieistotna lub myląca.

Poza tym aspektem, wykres 2.20 nie wykazywał żadnych niepokojących zachowań modelu. Cechy niosące teoretycznie najwięcej wartościowych informacji, pozostały wysoko w rankingu, a reszta parametrów praktycznie nie zmieniła wartości swojej wagi.



Rys. 2.20. Wykres wybranej grupy najważniejszych cech dla modelu 3

Zweryfikowano zatem ostatecznie ogólną skuteczność modelu 3, kompletując wcześniej wspomniane zestawienie, dostępne na Rys. 2.21. Do każdej tabeli porównującej rankingi dodano wartość funkcji *RBO*, odpowiadającą podobieństwu jego realnego rankingu do predyktowanego przez model. Miało to umożliwić obserwację oraz ocenę działania nowej funkcji celu, ponieważ tylko jej zastosowanie odróżnia nowy model od modelu drugiego.

TRAIN DATASET

2023 Real		2023 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.935	nikola jokić	0.816
shai gilgeous-alexander	0.646	luka dončić	0.459
luka dončić	0.572	shai gilgeous-alexander	0.427
giannis antetokounmpo	0.194	giannis antetokounmpo	0.366
jalen brunson	0.143	joel embiid	0.329
RBO Score: 0.8398			

2022 Real		2022 Predicted	
player	mvp votes share	player	pred
joel embiid	0.915	nikola jokić	0.594
nikola jokić	0.674	joel embiid	0.52
giannis antetokounmpo	0.606	stanley umude	0.342
jayson tatum	0.28	luka dončić	0.33
shai gilgeous-alexander	0.046	giannis antetokounmpo	0.293
RBO Score: 0.0000			

2011 Real		2011 Predicted	
player	mvp votes share	player	pred
lebron james	0.888	lebron james	0.565
kevin durant	0.735	kevin durant	0.224
chris paul	0.318	chris paul	0.209
kobe bryant	0.291	kyrylo fesenko	0.179
tony parker	0.274	kobe bryant	0.117
RBO Score: 0.9930			

2003 Real		2003 Predicted	
player	mvp votes share	player	pred
kevin garnett	0.991	kevin garnett	0.533
tim duncan	0.582	tim duncan	0.327
jermaine o'neal	0.425	bruno šundov	0.238
peja stojaković	0.228	tracy mcgrady	0.101
kobe bryant	0.172	ben wallace	0.093
RBO Score: 0.9661			

TEST DATASET

2021 Real		2021 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.875	nikola jokić	0.796
joel embiid	0.706	giannis antetokounmpo	0.583
giannis antetokounmpo	0.595	joel embiid	0.498
devin booker	0.216	luka dončić	0.227
luka dončić	0.146	lebron james	0.093
RBO Score: 0.8339			

2020 Real		2020 Predicted	
player	mvp votes share	player	pred
nikola jokić	0.961	nikola jokić	0.783
joel embiid	0.58	joel embiid	0.346
stephen curry	0.449	stephen curry	0.304
giannis antetokounmpo	0.345	giannis antetokounmpo	0.291
chris paul	0.138	luka dončić	0.159
RBO Score: 0.9989			

2019 Real		2019 Predicted	
player	mvp votes share	player	pred
giannis antetokounmpo	0.952	giannis antetokounmpo	0.64
lebron james	0.746	lebron james	0.498
james harden	0.363	james harden	0.415
luka dončić	0.198	luka dončić	0.23
kawhi leonard	0.166	damian lillard	0.129
RBO Score: 0.9989			

2018 Real		2018 Predicted	
player	mvp votes share	player	pred
giannis antetokounmpo	0.932	giannis antetokounmpo	0.79
james harden	0.768	james harden	0.695
paul george	0.352	nikola jokić	0.226
nikola jokić	0.21	paul george	0.182
stephen curry	0.173	anthony davis	0.103
RBO Score: 0.9661			

2017 Real		2017 Predicted	
player	mvp votes share	player	pred
james harden	0.955	james harden	0.846
lebron james	0.731	lebron james	0.529
anthony davis	0.441	anthony davis	0.265
damian lillard	0.205	damian lillard	0.168
russell westbrook	0.07	russell westbrook	0.117
RBO Score: 1.0000			

2016 Real		2016 Predicted	
player	mvp votes share	player	pred
russell westbrook	0.879	russell westbrook	0.706
james harden	0.746	james harden	0.637
kawhi leonard	0.495	kawhi leonard	0.495
lebron james	0.33	lebron james	0.302
isaiah thomas	0.08	john wall	0.155
RBO Score: 0.9689			

2015 Real		2015 Predicted	
player	mvp votes share	player	pred
stephen curry	1.0	stephen curry	0.874
kawhi leonard	0.484	lebron james	0.453
lebron james	0.482	russell westbrook	0.388
russell westbrook	0.371	kevin durant	0.366
kevin durant	0.112	kawhi leonard	0.34
RBO Score: 0.8339			

2014 Real		2014 Predicted	
player	mvp votes share	player	pred
stephen curry	0.922	stephen curry	0.701
james harden	0.72	james harden	0.635
lebron james	0.425	russell westbrook	0.305
russell westbrook	0.271	lebron james	0.233
anthony davis	0.156	chris paul	0.222
RBO Score: 0.9661			

2013 Real		2013 Predicted	
player	mvp votes share	player	pred
kevin durant	0.986	kevin durant	0.907
lebron james	0.713	lebron james	0.576
blake griffin	0.347	stephen curry	0.276
joakim noah	0.258	blake griffin	0.189
james harden	0.068	kevin love	0.168
RBO Score: 0.9661			

2012 Real		2012 Predicted	
player	mvp votes share	player	pred
lebron james	0.998	lebron james	0.945
kevin durant	0.632	kevin durant	0.765
carmelo anthony	0.393	carmelo anthony	0.314
chris paul	0.239	chris paul	0.257
kobe bryant	0.152	james harden	0.105
RBO Score: 0.9989			

2010 Real		2010 Predicted	
player	mvp votes share	player	pred
derrick rose	0.977	derrick rose	0.583
dwight howard	0.531	lebron james	0.502
lebron james	0.431	dwight howard	0.391
kobe bryant	0.354	kobe bryant	0.205
kevin durant	0.157	dwyane wade	0.13
RBO Score: 0.8398			

2009 Real		2009 Predicted	
player	mvp votes share	player	pred
lebron james	0.98	lebron james	0.976
kevin durant	0.495	kevin durant	0.436
kobe bryant	0.487	dwight howard	0.351
dwight howard	0.389	steve nash	0.264
dwyane wade	0.097	kobe bryant	0.187
RBO Score: 0.9661			

2008 Real		2008 Predicted	
player	mvp votes share	player	pred
lebron james	0.969	lebron james	0.905
kobe bryant	0.577	chris paul	0.506
dwyane wade	0.562	dwyane wade	0.482
dwight howard	0.271	kobe bryant	0.385
chris paul	0.159	dwight howard	0.243
RBO Score: 0.8608			

2007 Real		2007 Predicted	
player	mvp votes share	player	pred
kobe bryant	0.873	chris paul	0.6
chris paul	0.71	kobe bryant	0.391
kevin garnett	0.532	kevin garnett	0.362
lebron james	0.348	lebron james	0.315
dwight howard	0.048	dwight howard	0.191
RBO Score: 0.0339			

2006 Real		2006 Predicted	
player	mvp votes share	player	pred
dirk nowitzki	0.882	dirk nowitzki	0.777
steve nash	0.785	steve nash	0.601
kobe bryant	0.404	kobe bryant	0.317
tim duncan	0.222	tim duncan	0.268
lebron james	0.142	lebron james	0.209
RBO Score: 1.0000			

2005 Real		2005 Predicted	
player	mvp votes share	player	pred
steve nash	0.739	lebron james	0.471
lebron james	0.55	dirk nowitzki	0.464
dirk nowitzki	0.435	kobe bryant	0.371
kobe bryant	0.386	steve nash	0.295
chauncey billups	0.344	chauncey billups	0.283
RBO Score: 0.0011			

2004 Real		2004 Predicted	
player	mvp votes share	player	pred
steve nash	0.839	steve nash	0.574
shaquille o'neal	0.813	shaquille o'neal	0.438
dirk nowitzki	0.275	dirk nowitzki	0.239
tim duncan	0.258	lebron james	0.208
allen iverson	0.189	tim duncan	0.187
RBO Score: 0.9965			

2002 Real		2002 Predicted	
player	mvp votes share	player	pred
tim duncan	0.808	tim duncan	0.683
kevin garnett	0.732	kevin garnett	0.541
kobe bryant	0.417	tracy mcgrady	0.454
tracy mcgrady	0.359	kobe bryant	0.401
shaquille o'neal	0.106	dirk nowitzki	0.206
RBO Score: 0.9661			

2001 Real		2001 Predicted	
player	mvp votes share	player	pred
tim duncan	0.757	tim duncan	0.599
jason kidd	0.712	shaquille o'neal	0.4
shaquille o'neal	0.552	jason kidd	0.399
tracy mcgrady	0.31	tracy mcgrady	0.203
kobe bryant	0.078	allen iverson	0.118
RBO Score: 0.8398			

Rys. 2.21. Wyniki predykcji modelu 3

Podsumowując, model 3 osiągnął stosunkowo wysoką wartość średnią *RBO*, wynoszącą 0.8189, obliczoną na podstawie wyników uzyskanych z predykcji na zbiorze testowym i treningowym. Kluczowym osiągnięciem była jednak znacząca poprawa w precyzji przewidywania zawodnika *MVP*. Skuteczność prognozy najbardziej wartościowego zawodnika wzrosła z 73,9% do 86,9%, co przedstawiono w tabeli 2.5.

	Poprawnych predykcji	**Wszystkich predykcji**	**Skuteczność**
MVP	20	23	86.9%
Top 5 zawodników	94	115	81.7%

Tabela 2.5. Model 3 - tabela skuteczności predykcji

Osiągnięcie to odpowiada 3 dodatkowym poprawnym predykcjom w porównaniu do modelu poprzedniego, co pokazano na Rys 2.22

MVP Ranking in 2018-2019 season					
Real Ranking		**Model 2 Pred Ranking**		**Model 3 Pred Ranking**	
player	**mvp votes share**	**player**	**pred**	**player**	**pred**
giannis antetokounmpo	0.932	james harden	0.76	giannis antetokounmpo	0.79
james harden	0.768	giannis antetokounmpo	0.723	james harden	0.695
paul george	0.352	nikola jokić	0.259	nikola jokić	0.226
nikola jokić	0.21	paul george	0.24	paul george	0.182
stephen curry	0.173	stephen curry	0.133	anthony davis	0.103

MVP Ranking in 2016-2017 season					
Real Ranking		**Model 2 Pred Ranking**		**Model 3 Pred Ranking**	
player	**mvp votes share**	**player**	**pred**	**player**	**pred**
russell westbrook	0.879	james harden	0.695	russell westbrook	0.706
james harden	0.746	russell westbrook	0.67	james harden	0.637
kawhi leonard	0.495	kawhi leonard	0.456	kawhi leonard	0.495
lebron james	0.33	lebron james	0.301	lebron james	0.302
isaiah thomas	0.08	kevin durant	0.168	john wall	0.155

MVP Ranking in 2010-2011 season					
Real Ranking		**Model 2 Pred Ranking**		**Model 3 Pred Ranking**	
player	**mvp votes share**	**player**	**pred**	**player**	**pred**
derrick rose	0.977	lebron james	0.678	derrick rose	0.583
dwight howard	0.531	derrick rose	0.529	lebron james	0.502
lebron james	0.431	dwight howard	0.424	dwight howard	0.391
kobe bryant	0.354	kobe bryant	0.257	kobe bryant	0.205
kevin durant	0.157	dwyane wade	0.155	dwyane wade	0.13

Rys. 2.22. Poprawa prognozy *MVP* w sezonach 2010-11, 2016-17 i 2018-19

Pozostało tylko wyjaśnić kwestię trzech pomyłek modelu w prognozie *MVP*. Porównując jego błędy z błędami modeli poprzednich, zauważono że każdy z nich myli się w szczególnych dwóch sezonach (2005-06 ze zbioru treningowego oraz 2022-23 ze zbioru testowego).

Po wstępnej analizie wyników głosowań wymienionych sezonów odkryto zależność, która ostatecznie przesądziła o uznaniu modelu 3 za skuteczny. Wyniki prawdziwego głosowania z tamtych lat były i są do dziś uznawane za silnie kontrowersyjne w świecie koszykówki.

2005 Real		2005 Predicted	
player	mvp votes share	player	pred
steve nash	0.739	lebron james	0.471
lebron james	0.55	dirk nowitzki	0.464
dirk nowitzki	0.435	kobe bryant	0.371
kobe bryant	0.386	steve nash	0.295
chauncey billups	0.344	chauncey billups	0.283
RBO Score: 0.0011			

Rys. 2.23. Niepoprawna predykcja *MVP* w sezonie 2005-06

W sezonie 2005-06 na *MVP* wybrano przykładowo Steve'a Nash'a, notującego średnio 18.8 punktu, 10.5 asysty oraz 4.2 zbiórki na mecz. Choć były to imponujące liczby, jego rywalem w plebiscycie był między innymi Lebron James osiągający średnio 31 punktów, 6 asyst i 7 zbiórek, czy Shaquille O'Neal prezentujący równie wysokie wartości statystyk indywidualnych i dominujący fizycznie ponad innymi zawodnikami. Steve Nash został wybrany na *MVP* przez to, że prowadził i tworzył dynamiczny styl gry drużyny Phoenix Suns, który ostatecznie doprowadził ich do pierwszego miejsca w tabeli sezonu regularnego. Głosowanie zostało uznane za kontrowersyjne, ponieważ tak naprawdę ciężko ocenić, który z kandydatów był najbardziej wartościowym graczem, a o wyborze Steve'a Nash'a mogły zadecydować emocje głosujących ekspertów. Model 3 wybrał właśnie Lebron'a James'a na *MVP*, co nie jest ostatecznie dalekie od prawdy.

2022 Real		2022 Predicted	
player	mvp votes share	player	pred
joel embiid	0.915	nikola jokić	0.594
nikola jokić	0.674	joel embiid	0.52
giannis antetokounmpo	0.606	stanley umude	0.342
jayson tatum	0.28	luka dončić	0.33
shai gilgeous-alexander	0.046	giannis antetokounmpo	0.293
RBO Score: 0.0000			

Rys. 2.24. Niepoprawna predykcja *MVP* w sezonie 2022-23

W sezonie 2022 natomiast, pojedynek toczyli natomiast dwaj gracze, Joel Embid oraz Nikola Jokić.

W środowisku koszykarskim również mówi się o kontrowersji wyboru Joel'a Embida w tym sezonie. Wybór mógł wynikać z tego, że Nikola Jokić został wybrany najbardziej wartościowym graczem już w dwóch poprzednich sezonach. To skłoniło podobno głosujących, do wyboru nowego zawodnika, żeby zaprzeczyć wszelkim plotkom o ustawieniu wyników lub faworyzacji Jokić'a. Fakt, że model 3 prognozował Nikolę na *MVP*, nie powinien być zatem niepokojący. Nie odnotowano także istotnego spadku skuteczności w zakresie wyboru najlepszych pięciu zawodników sezonu. Model trzeci osiągnął trafność na poziomie 81,7%, podczas gdy model drugi charakteryzował się efektywnością wynoszącą 86% w tym obszarze. Ostatecznie zatem, kosztem niewielkiego obniżenia precyzji modelu w przewidywaniu pełnego rankingu sezonowego, uzyskano znaczną poprawę w podstawowym aspekcie projektu – prognozowaniu zawodnika *MVP*. Zmiana funkcji celu na dużo bardziej dopasowaną do tego specyficznego problemu i danych funkcję *RBO*, dała wyraźnie widoczne, pozytywne efekty. Model 3 można było zatem uznać za efektywny i przeprowadzić ostateczny test.

3. Predykcja MVP na sezon 2024-25

Ostatecznie sprawdzono model w jego docelowym zadaniu - prognozie *MVP* w sezonie, który jeszcze się nie zakończył. Sezon *NBA* 2024-25 rozpoczął się oficjalnie 23 października 2024 roku, a jego koniec przypada na 13 kwietnia 2025 roku. Ostateczny test modelu przeprowadzono natomiast 6 grudnia 2024 roku, co oznacza, że w dniu testu minęło zaledwie 44 z 173 dni sezonu regularnego. Na ten moment nie może być jeszcze pewności co do tego, który zawodnik wygra nagrodę *MVP*, ponieważ sezon jeszcze się nie zakończył. *NBA* publikuje natomiast co kilka tygodni ranking 3 faworytów do wygrania nagrody i to właśnie do niego porównano predykcję modelu.

Rys. 3.1. Oficjalny ranking faworytów do wygrania plebiscytu *MVP* w sezonie 2024-25, opublikowany przez *NBA* 6 grudnia 2024, źródło: Instagram NBA

2024-25 Predicted	
player	MVP Votes Share
nikola jokić	0.370
jt thor	0.353
giannis antetokounmpo	0.352

Rys. 3.2. Ranking 3 faworytów do wygrania plebiscytu *MVP* w sezonie 2024-25, prognozowany przez ostateczny model 6 grudnia 2024

Porównując oficjalny ranking dostępny na Rys 3.1 do prognozowanego przez model rankingu widocznego na Rys 3.2, od razu widać, że stworzone przeze mnie narzędzie poprawnie wybrało faworyta do wygrania plebiscytu *MVP* - w obydwu rankingach na 1 miejscu znajduje się bowiem Nikola Jokić. Dodatkowo, model uplasował Giannis'a Antetokounmpo na miejscu 3, podczas gdy w oficjalnym rankingu znajduje się on na bliskim predykcji miejscu 2. Niepokojące wydawać by się mogło, jak bliskie wartości *MVP Votes Share* otrzymali zawodnicy w rankingu prognozowanym. To zachowanie modelu nie jest jednak spowodowane żadnym błędem lub niedopatrzeniem, a wpływa na to raczej data wykonania testu. Każdy z zawodników, na dzień 6 grudnia, zdążył bowiem zagrać jedynie około 20 meczów. Dla tak małej próbki są zatem zebrane statystyki podawane na wejście modelu. Model był natomiast uczony na danych ze skończonych już sezonów, posiadających statystyki dla każdego zawodnika nawet z ponad 80 meczów. Biorąc pod uwagę, że test został wykonany w 44 dniu sezonu, można łatwo stwierdzić, że statystyki zawodników jeszcze nie zdążyły się ustabilizować, a próbka danych jest zbyt mała, żeby osiągnąć większe, dokładniejsze wartości *MVP Votes Share*. Mimo tych niedogodności, model dobrze prognozował 2 na 3 faworytów do nagrody *MVP* z oficjalnego rankingu, a co najważniejsze poprawnie przewidział faworyta do zostania najbardziej wartościowym zawodnikiem sezonu 2024-25.

4. Podsumowanie

Reasumując, udało się w pełni osiągnąć założony cel. Pomogły w tym systematyzacja procesu uczenia oraz wcześniejsze zapoznanie się ze stosowanymi metodami i algorytmami w problemach o zbliżonej tematyce. Kluczowe było też staranne przygotowanie i oczyszczenie zbioru danych. Wszystko to jednak nie miałoby znaczenia, gdyby nie konsekwentna analiza wyników, a następnie wprowadzanie kolejnych adaptacji. Implementacja i wykorzystanie funkcji *RBO*, czy też wprowadzenie nowego, własnego hiperparametru znacznie poprawiły bowiem skuteczności modelu. To właśnie one pozwoliły na złagodzenie efektów wysokiego niezbalansowania zbioru, które początkowo sprawiało modelowi trudności. Tak powstało narzędzie całkowicie spełniające swoją rolę - prognozujące zarówno najważniejszych kandydatów do wygrania plebiscytu, jak i skutecznie przewidujące kto powinien zostać najbardziej wartościowym zawodnikiem. Jak pokazują wyniki, narzędzie osiąga wysoką skuteczność w prognozach *MVP*. Przedstawiono jednak sytuacje, w których dodatkowym czynnikiem tworzącym ostateczny ranking i wpływającym na wybór *MVP* są ludzkie emocje. Ma to szczególne znaczenie, gdy wybór zostaje np. pomiędzy dwoma najlepszymi zawodnikami. Narzędzie nie posiada preferencji, a subiektywne zdanie głosujących nie zawsze pokryje się z bezstronną kalkulacją algorytmu. Nie da się zatem zawsze idealnie zasymulować przebiegu głosowania. Mimo tego, ostateczny test potwierdził, że zaimplementowane narzędzie jak najbardziej może sprawdzać się w prognozowaniu zwycięzcy plebiscytu *MVP*, zarówno po jak i dużo przed zakończeniem sezonu *NBA*.

Bibliografia

[1] Oficjalna strona ligi NBA, dostęp 6.12.2024. URL: *https://www.nba.com/*.

[2] P. Gabriel. „*Predicting the NBA MVP with Machine Learning*". Sept. 2022. URL: *https://towardsdatascience.com/predicting-the-nba-mvp-with-machine-learning-c3e5b755f42e*.

[3] Iqbal H. Sarker. „*Machine Learning: Algorithms, Real-World Applications and Research Directions*". W: t. 2. 3. Springer, 2021, s. 2–6. DOI: *10.1007/s42979-021-00592-x*.

[4] Josh Starmer. „*StatQuest: Random Forests Part 1 - Building, Using and Evaluating*". Dostęp: 15.09.2024. URL: *https://www.youtube.com/watch?v=J4Wdy0Wc_xQ*.

[5] Leo Breiman. „*Random Forests*". W: *Machine Learning*. T. 45. 1. Springer, 2001, s. 5–32. DOI: *10.1023/A:1010933404324*.

[6] Arnando Harlianto i Johan Setiawan. „*Forecasting the NBA's Most Valuable Player: A Regression Analysis Approach*". W: *Proceedings of the 2024 5th International Conference on Big Data Analytics and Practices (IBDAP)*. IEEE, 2024, s. 56–62. DOI: *10.1109/IBDAP62940.2024.10689693*.

[7] Mason Chen i Charles Chen. „*Data Mining Computing of Predicting NBA 2019–2020 Regular Season MVP Winner*". W: *Proceedings of the 2020 International Conference on Advances in Computing and Communication Engineering (ICACCE)*. IEEE, 2020, s. 1–5. DOI: *10.1109/ICACCE49060.2020.9155038*.

[8] Strona Basketball Reference, data dostępu: 6.12.2024. URL: *https://www.basketball-reference.com/*.

[9] Dokumentacja Random Forest Regressor, data dostępu: 29.10.2024. URL: *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor*.

[10] Sandeep R. „*Mastering Random Forests: A comprehensive guide*". Oct. 2020. URL: *https://towardsdatascience.com/mastering-random-forests-a-comprehensive-guide-51307c129cb1*.

[11] William Webber, Alistair Moffat i Justin Zobel. „*A similarity measure for indefinite rankings*". W: *ACM Trans. Inf. Syst.* 28.4 (list. 2010). ISSN: 1046-8188. DOI: *10.1145/1852102.1852106*.

Dodatek 1

Tabela 1. Tabela średnich statystyk na mecz

Skrót	**Pełna nazwa**	**Opis**
Rk	*Rank*	Id gracza w tabeli
Age	*Player's Age on February 1 of the season*	Wiek gracza 1 lutego
Pos	*Position*	Pozycja gracza (PG, SG, SF, PF, C)
G	*Games*	Liczba rozegranych meczów przez gracza w sezonie
GS	*Games Started*	Liczba meczów, w których gracz wystartował w pierwszym składzie
MP	*Minutes Played*	Liczba minut spędzonych przez gracza na boisku na mecz
FG	*Field Goals*	Średnia liczba celnych rzutów z gry na mecz
FGA	*Field Goal Attempts*	Średnia liczba prób rzutów z gry na mecz
FG%	*Field Goal Percentage*	Średni procent celnych rzutów z gry na mecz
3P	*3-Point Field Goals*	Średnia liczba celnych rzutów za 3 punkty na mecz
3PA	*3-Point Field Goal Attempts*	Średnia liczba prób rzutów za 3 punkty na mecz
3P%	*3-Point Field Goal Percentage*	Średni procent celnych rzutów za 3 punkty na mecz
2P	*2-Point Field Goals*	Liczba trafionych rzutów za 2 punkty na mecz
2PA	*2-Point Field Goal Attempts*	Średnia liczba prób rzutów za 2 punkty na mecz
2P%	*2-Point Field Goal Percentage*	Średnia liczba celnych rzutów za 2 punkty na mecz
EFG%	*Effective Field Goal Percentage*	Średnia skuteczność rzutów z uwzględnieniem różnic w wartości rzutów 3pkt i 2 pkt
FT	*Free Throws*	Średnia liczba celnych rzutów osobistych na mecz
FTA	*Free Throw Attempts*	Średnia liczba prób rzutów osobistych na mecz
FT%	*Free Throw Percentage*	Średni procent celnych rzutów osobistych na mecz
ORB	*Offensive Rebounds*	Średnia liczba zbiórek ofensywnych na mecz
DRB	*Defensive Rebounds*	Średnia liczba zbiórek defensywnych na mecz
TRB	*Total Rebounds*	Średnia liczba wszystkich zbiórek na mecz
AST	*Assists*	Średnia liczba asyst na mecz
STL	*Steals*	Średnia liczba przechwytów na mecz
BLK	*Blocks*	Średnia liczba bloków na mecz
TOV	*Turnovers*	Średnia liczba strat na mecz
PF	*Personal Fouls*	Średnia liczba fauli osobistych na mecz
PTS	*Points*	Średnia liczba zdobytych punktów na mecz

Tabela 2. Tabela statystyk zaawansowanych

Skrót	**Pełna nazwa**	**Opis**
Rk	*Rank*	Id gracza w tabeli
Age	*Player's Age on February 1 of the season*	Wiek gracza
Tm	*Team*	Drużyna, w której gra gracz
G	*Games*	Łączna liczba gier zagranych sezonie
MP	*Minutes Played*	Łączna Liczba minut spędzonych przez gracza na boisku
PER	*Player Efficiency Rating*	Miara produktywności gracza na minutę, średnia ligowa to 15
TS%	*True Shooting Percentage*	Miara skuteczności rzutów 2- i 3-punktowych oraz rzutów osobistych
3PAr	*3-Point Attempt Rate*	Procent rzutów z pola, które są rzutami za 3 punkty
FTr	*Free Throw Attempt Rate*	Liczba prób rzutów osobistych na rzut z pola
ORB%	*Offensive Rebound Percentage*	Estymowana ilość zbiórek ofensywnych, które gracz zebrał, w porównaniu do możliwych do zebrania
DRB%	*Defensive Rebound Percentage*	Estymowana ilość zbiórek defensywnych, które gracz zebrał, w porównaniu do możliwych do zebrania
TRB%	*Total Rebound Percentage*	Estymowana ilość zbiórek ogólnych, które gracz zebrał, w porównaniu do możliwych do zebrania
AST%	*Assist Percentage*	Estymowany procent skuteczności asyst, gdy gracz był na boisku
STL%	*Steal Percentage*	Estymowana skuteczność przejęć, gdy gracz był na boisku
BLK%	*Block Percentage*	Estymowana skuteczność bloków, gdy gracz był na boisku
TOV%	*Turnover Percentage*	Estymowana ilość strat w których uczestniczył gracz, na 100 zagrań drużyny
USG%	*Usage Percentage*	Estymowany procent akcji drużyny, w których uczestniczył gracz
OWS	*Offensive Win Shares*	Estymowana ilość zwycięstw gracza dzięki jego ofensywie
DWS	*Defensive Win Shares*	Estymowana ilość wygranych meczów przez gracza dzięki jego defensywie
WS	*Win Shares*	Estymowana ilość zwycięstw, na które gracz miał wpływ
WS/48	*Win Shares Per 48 Minutes*	Estymowana ilość zwycięstw na 48 minut (średnia ligowa to 0.1)
OBPM	*Offensive Box Plus/Minus*	Szacunkowa liczba punktów ofensywnych na 100 posiadań drużyny, powyżej przeciętnego gracza
DBPM	*Defensive Box Plus/Minus*	Szacunkowa liczba punktów defensywnych na 100 posiadań drużyny, powyżej przeciętnego gracza
BPM	*Box Plus/Minus*	Szacunkowa liczba punktów na 100 posiadań drużyny, powyżej przeciętnego gracza
VORP	*Value over Replacement Player*	Szacunkowa liczba punktów na 100 posiadań drużyny, które gracz wniósł ponad poziom zastępczy (zmiennika)

Tabela 3. Tabela statystyk sumarycznych

Skrót	**Pełna nazwa**	**Opis**
Rk	*Rank*	Id gracza w tabeli
Age	*Player's Age on February 1 of the season*	Wiek gracza
Pos	*Position*	Pozycja gracza
GS	*Games Started*	Łączna liczba meczów, w których zawodnik wystartował w pierwszym składzie
MP	*Minutes Played*	Łączna liczba minut spędzonych przez gracza na boisku w całym sezonie
FG	*Field Goals*	Łączna liczba celnych rzutów z gry
FGA	*Field Goal Attempts*	Łączna liczba prób rzutów z gry
FG%	*Field Goal Percentage*	Łączny procent celnych rzutów z gry
3P	*3-Point Field Goals*	Łączna liczba celnych rzutów za 3 punkty
3PA	*3-Point Field Goal Attempts*	Łączna liczba prób rzutów za 3 punkty
3P%	*3-Point Field Goal Percentage*	Łączny procent celnych rzutów za 3 punkty w stosunku do prób
2P	*2-Point Field Goals*	Łączna liczba celnych rzutów za 2 punkty
2PA	*2-Point Field Goal Attempts*	Łączna liczba prób rzutów za 2 punkty
2P%	*2-Point Field Goal Percentage*	Łączny procent celnych rzutów za 2 punkty w stosunku do prób
EFG%	*Effective Field Goal Percentage*	Łączna skuteczność rzutów z uwzględnieniem różnic w wartości rzutów 3pkt i 2 pkt
FT	*Free Throws*	Łączna liczba celnych rzutów osobistych
FTA	*Free Throw Attempts*	Łączna liczba prób rzutów osobistych
FT%	*Free Throw Percentage*	Procent celnych rzutów osobistych w stosunku do prób
ORB	*Offensive Rebounds*	Łączna liczba zbiórek ofensywnych
DRB	*Defensive Rebounds*	Łączna liczba zbiórek defensywnych
TRB	*Total Rebounds*	Łączna liczba wszystkich zbiórek
AST	*Assists*	Łączna liczba asyst
STL	*Steals*	Łączna liczba przechwytów
BLK	*Blocks*	Łączna liczba bloków
TOV	*Turnovers*	Łączna liczba strat
PF	*Personal Fouls*	Łączna liczba zdobytych punktów
PTS	*Points*	Łączna liczba zdobytych punktów

Tabela 4. Tabela klasyfikacji drużyn

Skrót	**Pełna nazwa**	**Opis**
W	*Wins*	Liczba zwycięstw drużyny w sezonie
L	*Losses*	Liczba porażek drużyny w sezonie
W/L%	*Winning Percentage*	Procent zwycięstw drużyny w stosunku do rozegranych meczów
GB	*Games Behind*	Liczba meczów, o które drużyna jest w tyle za liderem w tabeli
PS/G	*Points Scored Per Game*	Średnia liczba punktów zdobywanych przez drużynę na mecz
PA/G	*Points Allowed Per Game*	Średnia liczba punktów traconych przez drużynę na mecz
SRS	*Simple Rating System*	Prosta ocena zespołu uwzględniająca różnicę punktów w meczach oraz siłę przeciwników

Dodatek 2

Tabela 5. Ostateczny zbiór statystyk

Skrót	**Rozwinięcie**	**Opis**
player	*Player first and second name*	Imie i nazwisko gracza
team	*Players team*	Drużyna w której gra gracz
gs_total	*Total Games started*	Liczba meczów, w których gracz wystartował w pierwszym składzie
fg_total	*Total Field Goals*	Łączna liczba celnych rzutów z gry
fga_total	*Total Field Goal Attempts*	Łączna liczba prób rzutów z gry
fg%_total	*Total Field Goal Percentage*	Łączny procent celnych rzutów z gry
3p_total	*Total 3-Point Field Goals*	Łączna liczba celnych rzutów za 3 punkty
3pa_total	*Total 3-Point Field Goal Attempts*	Łączna liczba prób rzutów za 3 punkty
3p%_total	*Total 3-Point Field Goal Percentage*	Łączny procent celnych rzutów za 3 punkty w stosunku do prób
2p_total	*Total 2-Point Field Goals*	Łączna liczba celnych rzutów za 2 punkty
2pa_total	*Total 2-Point Field Goal Attempts*	Łączna liczba prób rzutów za 2 punkty
2p%_total	*Total 2-Point Field Goal Percentage*	Łączny procent celnych rzutów za 2 punkty w stosunku do prób
efg%_total	*Total Effective Field Goal Percentage*	Łączna skuteczność rzutów z uwzględnieniem różnic w wartości rzutów 3pkt i 2 pkt
ft_total	*Total Free Throws*	Łączna liczba celnych rzutów osobistych
fta_total	*Total Free Throw Attempts*	Łączna liczba prób rzutów osobistych
ft%_total	*Total Free Throw Percentage*	Procent celnych rzutów osobistych w stosunku do prób
orb_total	*Total Offensive Rebounds*	Łączna liczba zbiórek ofensywnych
drb_total	*Total Defensive Rebounds*	Łączna liczba zbiórek defensywnych
trb_total	*Total Rebounds*	Łączna liczba wszystkich zbiórek
ast_total	*Total Assists*	Łączna liczba asyst
stl_total	*Total Steals*	Łączna liczba przechwytów
blk_total	*Total Blocks*	Łączna liczba bloków
tov_total	*Total Turnovers*	Łączna liczba strat
pf_total	*Total Personal Fouls*	Łączna liczba zdobytych punktów
pts_total	*Total Points*	Łączna liczba zdobytych punktów
mp_per_game	*Minutes Played Per Game*	Liczba minut spędzonych przez gracza na boisku na mecz
fg_per_game	*Field Goals Per Game*	Średnia liczba celnych rzutów z gry na mecz
fga_per_game	*Field Goal Attempts Per Game*	Średnia liczba prób rzutów z gry na mecz
fg%_per_game	*Field Goal Percentage Per Game*	Średni procent celnych rzutów z gry na mecz
3p_per_game	*3-Point Field Goals Per Game*	Średnia liczba celnych rzutów za 3 punkty na mecz
3pa_per_game	*3-Point Field Goal Attempts Per Game*	Średnia liczba prób rzutów za 3 punkty na mecz

3p%_per_game	*3-Point Field Goal Percentage Per Game*	Średni procent celnych rzutów za 3 punkty na mecz
2p_per_game	*2-Point Field Goals Per Game*	Liczba trafionych rzutów za 2 punkty na mecz
2pa_per_game	*2-Point Field Goal Attempts Per Game*	Średnia liczba prób rzutów za 2 punkty na mecz
2p%_per_game	*2-Point Field Goal Percentage Per Game*	Średnia liczba celnych rzutów za 2 punkty na mecz
efg%_per_game	*Effective Field Goal Percentage Per Game*	Średnia skuteczność rzutów z uwzględnieniem różnic w wartości rzutów 3pkt i 2 pkt
ft_per_game	*Free Throws Per Game*	Średnia liczba celnych rzutów osobistych na mecz
fta_per_game	*Free Throw Attempts Per Game*	Średnia liczba prób rzutów osobistych na mecz
ft%_per_game	*Free Throw Percentage Per Game*	Średni procent celnych rzutów osobistych na mecz
orb_per_game	*Offensive Rebounds Per Game*	Średnia liczba zbiórek ofensywnych na mecz
drb_per_game	*Defensive Rebounds Per Game*	Średnia liczba zbiórek defensywnych na mecz
trb_per_game	*Total Rebounds Per Game*	Średnia liczba wszystkich zbiórek na mecz
ast_per_game	*Assists Per Game*	Średnia liczba asyst na mecz
stl_per_game	*Steals Per Game*	Średnia liczba przechwytów na mecz
blk_per_game	*Blocks Per Game*	Średnia liczba bloków na mecz
tov_per_game	*Turnovers Per Game*	Średnia liczba strat na mecz
pf_per_game	*Personal Fouls Per Game*	Średnia liczba fauli osobistych na mecz
pts_per_game	*Points Per Game*	Średnia liczba zdobytych punktów na mecz
age	*Player's Age on February 1 of the season*	Wiek gracza
g	*Games*	Łączna liczba gier zagranych sezonie
mp	*Minutes Played*	Łączna Liczba minut spędzonych przez gracza na boisku
per	*Player Efficiency Rating*	Miara produktywności gracza na minutę, średnia ligowa to 15
ts%	*True Shooting Percentage*	Miara skuteczności rzutów 2- i 3-punktowych oraz rzutów osobistych
3par	*3-Point Attempt Rate*	Procent rzutów z pola, które są rzutami za 3 punkty
ftr	*Free Throw Attempt Rate*	Liczba prób rzutów osobistych na rzut z pola
orb%	*Offensive Rebound Percentage*	Estymowana ilość zbiórek ofensywnych, które gracz zebrał, w porównaniu do możliwych do zebrania
drb%	*Defensive Rebound Percentage*	Estymowana ilość zbiórek defensywnych, które gracz zebrał, w porównaniu do możliwych do zebrania
trb%	*Total Rebound Percentage*	Estymowana ilość zbiórek ogólnych, które gracz zebrał, w porównaniu do możliwych do zebrania
ast%	*Assist Percentage*	Estymowany procent skuteczności asyst, gdy gracz był na boisku
stl%	*Steal Percentage*	Estymowana skuteczność przejęć, gdy gracz był na boisku
blk%	*Block Percentage*	Estymowana skuteczność bloków, gdy gracz był na boisku

tov%	*Turnover Percentage*	Estymowana ilość strat w których uczestniczył gracz, na 100 zagrań drużyny
usg%	*Usage Percentage*	Estymowany procent akcji drużyny, w których uczestniczył gracz
ows	*Offensive Win Shares*	Estymowana ilość zwycięstw gracza dzięki jego ofensywie
dws	*Defensive Win Shares*	Estymowana ilość wygranych meczów przez gracza dzięki jego defensywie
ws	*Win Shares*	Estymowana ilość zwycięstw, na które gracz miał wpływ
ws/48	*Win Shares Per 48 Minutes*	Estymowana ilość zwycięstw na 48 minut (średnia ligowa to 0.1)
obpm	*Offensive Box Plus/Minus*	Szacunkowa liczba punktów ofensywnych na 100 posiadań drużyny, powyżej przeciętnego gracza
dbpm	*Defensive Box Plus/Minus*	Szacunkowa liczba punktów defensywnych na 100 posiadań drużyny, powyżej przeciętnego gracza
bpm	*Box Plus/Minus*	Szacunkowa liczba punktów na 100 posiadań drużyny, powyżej przeciętnego gracza
vorp	*Value over Replacement Player*	Szacunkowa liczba punktów na 100 posiadań drużyny, które gracz wniósł ponad poziom zastępczy (zmiennika)
seed	*Ranking position of the team*	Miejsce drużyny zawodnika w tabeli wynikające z bilansu wygranych i przegranych
mvp votes share	*share of votes for the player in MVP voting*	Udział głosów oddanych na gracza w głosowaniu
season	*season labeled by year*	Sezon w którym zostały zebrane statystyki