Przewidywanie wartości transferowych piłkarzy

Uladzislau Partnou

5 czerwca, 2023

Treść

1 Opis problemu

Problem, który będę rozważać, dotyczy przewidywania wartości transferowych zawodników piłki nożnej. Transferowe wartości zawodników są niezwykle istotne w dzisiejszym świecie piłki nożnej, gdzie kwoty przeprowadzanych transferów sięgają dziesięciu i nawet stu milionów dolarów. Przewidywanie tych wartości ma duże znaczenie dla klubów, agencji piłkarskich i innych podmiotów zainteresowanych tym rynkiem.

Celem problemu jest stworzenie modelu prognostycznego, który będzie mógł oszacować wartość transferową danego zawodnika na podstawie różnych czynników, takich jak wiek, pozycja piłkarza, statystyka, za jaki klub występuje itp.

Przewidywanie wartości transferowych zawodników piłki nożnej może przyczynić się do bardziej efektywnego zarządzania zasobami klubów piłkarskich, lepszej oceny inwestycji w talenty, a także pomóc w odkrywaniu młodych utalentowanych zawodników.

2 Zbieranie danych

Aby przeprowadzić analizę wartości transferowych zawodników piłki nożnej oraz innych czynników z nimi związanych, istnieje wiele źródeł danych, które można wykorzystać. Dwa popularne portale internetowe, które dostarczają szeroki zakres statystyk piłkarskich i informacji o transferach, to Transfermarkt oraz FBREF.

Dla przechowywania datasetów w projekcie użyłem klasę DataFrame z biblioteki pandas

2.1 FBREF

FBref to platforma, która dostarcza zaawansowanych statystyk piłkarskich. FBref oferuje bogate źródło danych dotyczących drużyn, zawodników, meczów, osiągnięć i wielu innych czynników związanych z piłką nożną.

Zdecydowałem, że będę pobierał dane tylko z top 5 Lig (angielska Premier League, hiszpańska La Liga, włoska Serie A, niemiecka Bundesliga i francuska Ligue 1), ponieważ dysponują najbardziej kompleksowymi danymi.

Podjęłem decyzję o pobieraniu danych z ostatnich trzech sezonów, ponieważ wierzę, że taki okres analizy pozwoli nam na bardziej precyzyjne przewidywanie wartości transferowych piłkarzy. Istnieją sytuacje, w których zawodnik w dwóch poprzednich sezonach prezentował się doskonale, co skutkowało wzrostem jego wartości transferowej. Jednak w ostatnim sezonie jego występy były słabe, co spowodowało spadek jego wartości. W takim przypadku wartość transferowa tego piłkarza nie będzie równa wartości transferowej innego zawodnika, który ma podobną statystykę, ale w dwóch poprzednich sezonach prezentował się gorzej. Analiza danych z ostatnich trzech sezonów pozwoli nam uwzględnić takie sytuacje i uzyskać bardziej kompleksowy obraz wartości transferowych piłkarzy. Biorąc pod uwagę występy z większej ilości sezonów, możemy spojrzeć na długoterminowe trendy i ewolucję w grze zawodników. Ta informacja jest niezwykle istotna, ponieważ pokazuje nam, czy dany piłkarz utrzymuje swoją dobrą formę przez dłuższy okres czasu.

2.1.1 Opis różnych statystyk

Dla każdej ligi istnieje 10 różnych zestawów danych mierzących różne aspekty gry zawodnika:

- Standardowa statystyka (Standard Stats)
- Bramkarstwo (Goalkeeping)
- Zaawansowane bramkarstwo (Advanced Goalkeeping)
- Strzelanie (Shooting)
- Podania (Passing)
- Rodzaje podań (Pass Types)
- Tworzenie goli i strzałów (Goal and Shot Creation)
- Działania defensywne (Defensive Actions)
- Posiadanie piłki (Possession)
- Różne statystyki (Miscellaneous Stats)

Biorąc pod uwagę, że bramkarze są oceniani na podstawie zupełnie innych wskaźników w porównaniu z zawodnikami z pola, podjęłem decyzję o nieuwzględnianiu bramkarzy w tym projekcie. W związku z tym dwa zbiory danych dotyczące występów bramkarzy zostały pominięte.

Przegląd pozostałych zestawów danych:

- Standardowa statystyka: standardowa informacja o wieku każdego zawodnika, czasie gry, strzelone gole i asysty, oczekiwane gole i asysty, liczba żółtych/czerwonych kartek itp.
- Strzały: Informacje dotyczące strzałów zawodników z ilościowego i jakościowego punktu widzenia.
- Podania: Informacje dotyczące ilości i jakości podań, podzielone na sekcje w oparciu o odległość podania (krótkie, średnie i długie).

- Rodzaje podań: Informacje dotyczące rodzaju wykonywanych podań i ich rezultatów (tj. wysokość z powietrza/średni poziom/ziemia i część ciała użyta do wykonania podania).

  Zdecydowałem nie pobierać dane dotyczące rodzajów podań, bo moim zdaniem części ciała używane przez gracza do wykonania podania lub wysokość, na której gracze wykonują podania, prawdopodobnie nie mają wpływ na cenę gracza.

- Tworzenie bramek i strzałów: Informacje dotyczące działań piłkarzy, które doprowadziły do możliwości oddania strzału i zdobycia bramki.

- Działania defensywne: Informacje dotyczące defensywnych aspektów gry piłkarza, a także informacje o tym, w jaki sposób jego wysiłki w obronie przyczyniły się do odzyskania piłki przez drużynę i stworzenia w rezultacie okazji do zdobycia bramki.

- Posiadanie piłki: Informacje dotyczące zdolności piłkarza do prowadzenia piłki i wpływania na przebieg gry.

- Różne statystyki: Różne informacje dotyczące występów na boisku, takie jak liczba bezpośrednich czerwonych kartek, drugich żółtych kartek, popełnionych fauli, ofsajdów itp.

  Jak i w przypadku z rodzajem podań, zdecydowałem nie pobierać ten rodzaj statystyki, ponieważ moim zdaniem nie ma on wpływu na cenę gracza

W końcu wyszło, że dane, które będą pobrane z FBREF, to:

- Standardowa statystyka
- Strzały
- Podania
- Tworzenie bramek i strzałów
- Działania defensywne
- Posiadanie piłki

2.1.2 Pobieranie danych

Postanowiłem najpierw pobrać tabelki ze statystykami za pomocą przycisku "Get as Excel Workbook", jednak napotkałem ograniczenie, ponieważ maksymalna liczba wierszy wynosiła 500, a liczba piłkarzy przekraczała tę wartość (około 600). W związku z tym, zdecydowałem się skorzystać z innego przycisku, "Get Table as CSV", który generował tekstowe reprezentacje tabel. Następnie każdy ten tekst wkleiłem do Excelu i zapisałem plik do projektu. W ten sposób udało mi się uzyskać 90 tabel (3 sezony * 5 lig * 6 rodzajów statystyk).

Podczas analizy tych tabel zauważyłem, że niektóre nazwy kolumn się powtarzały, mimo że miały różne znaczenia. Na przykład, w tabeli dotyczącej podań, te same nazwy kolumn były używane dla podań krótkich, średnich i długich. Ponadto, większość nazw kolumn była nieczytelna i trudna do zrozumienia. W związku z tym, postanowiłem ręcznie zmienić nazwy kolumn dla wszystkich tabel dotyczących statystyk z sezonu 22/23 w angielskiej Premier League. Te nowe nazwy kolumn zostały użyte jako wzór do zmiany nazw kolumn w pozostałych tabelach.

2.1.3 Połączenie wszystkich danych z 3 sezonów w jeden dataset

Przy połączeniu datasetów trzeba było uwzględnić kilka rzeczy:

- Piłkarzy, którzy przeszli w połowie sezonu do innej drużyny, pojawiali się dwukrotnie. Aby rozwiązać ten problem, dla każdego rodzjau statystyki utworzono funkcję, która agregowała dwa wiersze dla tych piłkarzy w jeden.
- Niektóre nazwy piłkarzy na stronach FBREF i Transfermarkt różniły się od siebie. W związku z tym, stworzyłem funkcję, która zamienia litery w nazwie piłkarza spoza angielskiego alfabetu na ich odpowiedniki w alfabecie angielskim.
- Niektóre kolumny w datasetach się duplikowały(np. za jaki klub dany piłkarz występuje). Oprócz tego, istniały kolumny z informacją, którą pobierałem z Transfermarktu. Dlatego te kolumny były usuniętę za pomocą funkcji drop.

Proces połączenia wyglądał następująco:

1. Przeiterowałem po krajach, sezonach i rodzajach statystyki, pobierając dane z plików i przechowując je w słowniku.

2. Zrobiłem słownik lambda funkcji, każda z których przetwarza dataset z odpowiednim rodzajem statystyki tak, żeby nie było żadnych powtarzających kolumn oraz nazw piłkarzy z literami spoza angielskiego alfabetu

3. Połączyłem datasety po krajach za pomocą pandas.concat

4. Użyłem odpowiednie lambdy do przetworzenia datasetów dla każdej statystyki, które ułożyłem do listy.

5. Połączyłem tę listę datasetów w DataFrame. Stworzyłem dla tego celu funkcję, która przyjmuje listę datasetów oraz numer sezonu. Najpierw łączy ona datasety za pomocą funkcji reduce z biblioteki functools oraz pandas.merge, a następnia zmienia nazwy kolumn, dodając do każdej kolumny numer sezonu, do którego cecha, którą ta kolumna reprezentuje, się odnosi.

6. Po wszystkich tych operacjach pozostało się 3 datasety, reprezentujące każdy sezon, które połączyłem w jeden DataFrame za pomocą pd.merge z użyciem parametru how='outer'. To powoduje, że zamiast porzucania wierszy, które nie pojawiają się we wszystkich datasetach, wszystkim atrybutom wierszu, który nie pojawił się w datasecie, przypisują się wartości nan.

Wreszcie otrzymałem jeden Dataframe, który zawiera wszystką statystykę piłkarzy z top 5 lig w ostatnich 3 sezonach.

Do zapisywania datasetu do pliku z rozszerzeniem .xlsx zrobiłem funkcję, która przyjmuje zapisywany Dataframe oraz nazwę pliku. Funkcja ta najpierw tworzy całą ścieżkę do pliku za pomocą funkcji getcwd oraz path.join z modułu os. Funkcja getcwd pobiera bieżącą ścieżkę, a funkcja path.join tworzy całą ścieżkę do pliku wraz z nazwą pliku. Dataframe zostaje zapisywany do pliku za pomocą funkcji toexcel klasy DataFrame.

Dataset zawierający dane piłkarzy ze strony FBREF zapisałem do pliku FBREF.xlsx

2.2 Transfermarkt

Transfermarkt to platforma internetowa, która posiada następującą informację o piłkarzach:

- Imię i nazwisko
- Wiek
- Narodowość
- Pozycja
- Miejsce urodzenia
- Wzrost
- Preferowana stopa
- Data dołączenia do obecnego zespołu
- Agent piłkarza
- Obecny klub
- Data wygaśnięcia kontraktu
- Sponsor
- Aktualna wartość transferowa w euro (zmienna docelowa)

Przyjęłem decyzję, że nie będę pobierał dane dotyczące agenta, miejsca urodzenia, datę dołączenia do obecnego zespołu oraz sponsoru piłkarza, ponieważ prawdopodobnie nie wpływają one na wartość transferową piłkarza oraz nie jest to informacja która mi interesuje. Też zdecydowałem, że będę pobierał dane piłkarzy z bieżącego sezonu, ponieważ chciałem mieć aktualną informację o każdym piłkarze.

W końcu wyszło, że dane piłkarza, które będą pobrane z Transfermarktu, to:

- Imię i nazwisko
- Wiek

- Narodowość
- Pozycja
- Preferowana stopa
- Wzrost
- Obecny klub
- Data wygaśnięcia kontraktu
- Aktualna wartość transferowa

2.2.1 Pobieranie danych

Do pobierania danych z Transfermarktu wykorzystałem 2 biblioteki: requests oraz BeautifulSoup. Za pomocą biblioteki requests Wykonywałem zapytania do stron, a za pomocą BeautifulSoup analizowałem HTML kod tych stron, żeby wyciągnąć wymaganą informację.

Dla scrapowania danych piłkarzy zrobiłem 2 funkcji:

Pierwsza funkcja pobiera i zwraca linki do wszystkich piłkarzy z top 5 lig. Ta funkcja najpierw przechodzi po stronach top 5 lig, pobierając linki do wszystkich klubów występujących w danych ligach. Ogólnie takich klubów jest 98 (Bundesliga ma 18 klubów, pozostałe 4 top ligi - 20 klubów). Potem ta funkcja przechodzi po stronie każdego klubu i pobiera linki do piłkarzy występujących w danym klubie.

Druga funkcja przyjmuje linki do stron piłkarzy, przechodzi po stronie każdego piłkarza i pobiera wymaganą informację. Robi się to za pomocą funkcji find klasy BeautifulSoup.

Zamiast tego, żeby przechowywać informację o dacie wygaśnięcia kontraktu, zdecydowałem przechowywać ilość roków, pozostałych do wygaśnięcia kontraktu. Jeśli strona piłkarza nie zawierała informację o dacie wygaśnięcia kontraktu, przyjąłem, że jego kontrakt wygasa w tym roku.

Przy scrapowaniu stron niektórych piłkarzy, nie wszystkie wymagane dane były znalezione. Podjęłem decyzję nie dodawać do datasetu takich piłkarzy.

Po zebraniu wszystkich danych w jeden DataFrame, użyłem wcześniej omówionej funkcji, która zmienia nazwy piłkarzy tak, aby odpowiadały angielskiemu alfabetowi

W końcu miałem dataset ze wszystkimi danymi piłkarzy ze strony Transfermarkt, który zapisałem do pliku transfermarkt.xlsx

2.3 Łączenie danych FBREF i Transfermarkt

Przy łączeniu danych FBREF i Transfermarkt wykorzystałem funkcję pd.merge z parametrem how='inner', bo chciałem, żeby łączny Dataframe zawierał tylko piłkarzy, którzy są w obu datasetach. Łączny DataFrame został zapisany do pliku fbrefTransfermarktDataset.xlsx

3 Wyczyszczenie danych

W tym momencie miałem cały dataset, ale przed tym jak wykorzystywać go do tworzenia modeli, trzeba było zrobić kilka rzeczy.

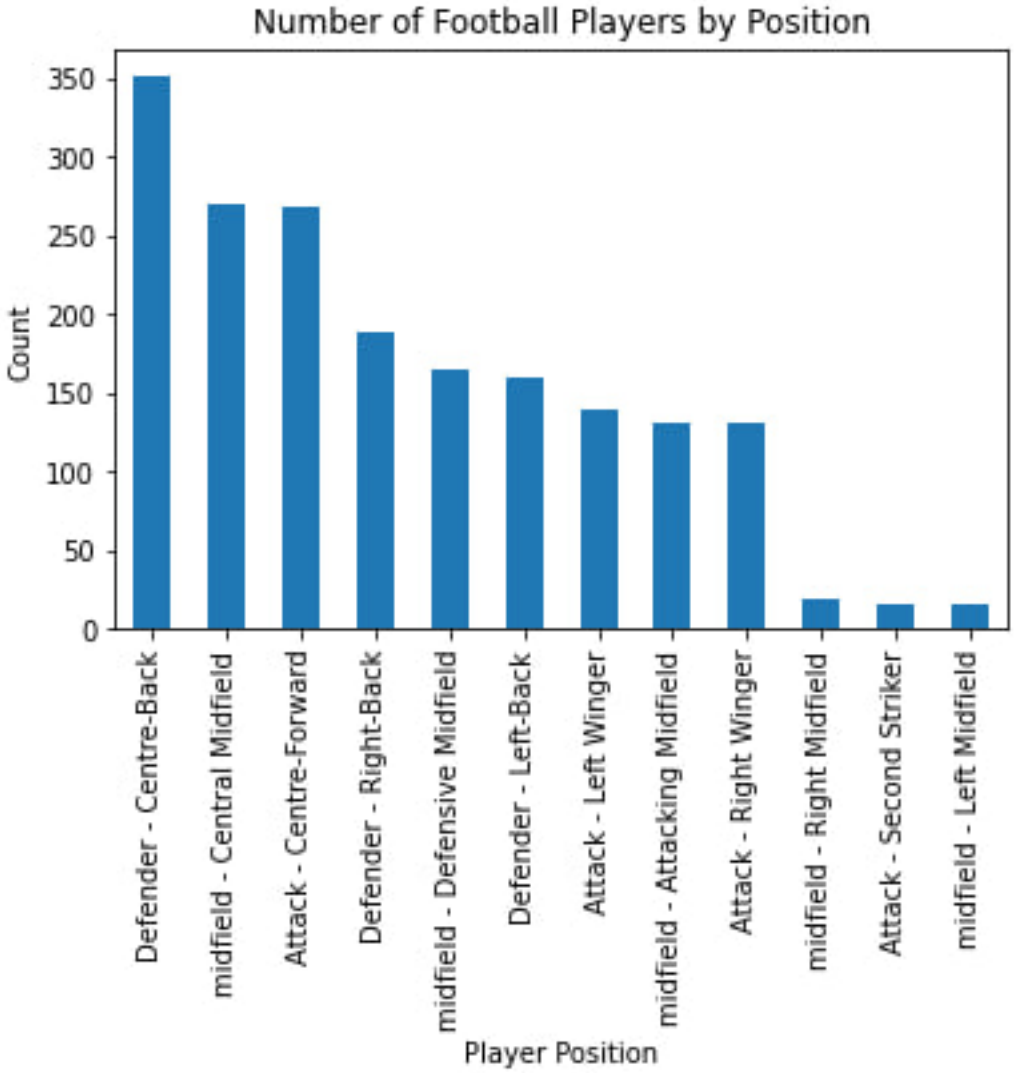
Po pierwze, ten dataset zawierał bramkarzy, wartość transferową których nie chce przewidywać. W tym celu zrobiłem funkcję, która odrzuca bramkarzy.

Po drugie, trzeba było zamienić nan wartości na coś sensowne. W tym celu zrobiłem funkcję, która bierze średnie wartości dla każdego wskaźnika z sezonów, w których piłkarz rywalizował w 5 najlepszych ligach i zastępuje wartości nan tymi wartościami.

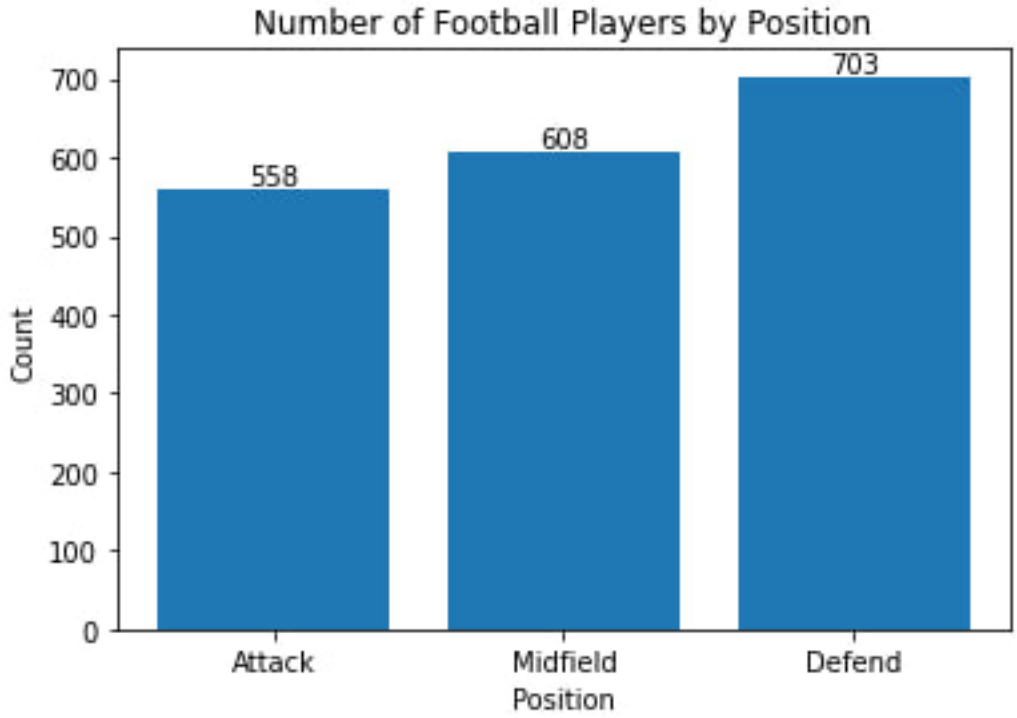
W końcu miałem gotowy Dataframe, który był gotowy do tego, żeby tworzyć modele przewidujące. Finalny dataset zapisałem do pliku finalDataset.xlsx. Ten dataset zawiera informację o 1870 piłkarzy.

4 Wstępna analiza oraz wizualizacja danych

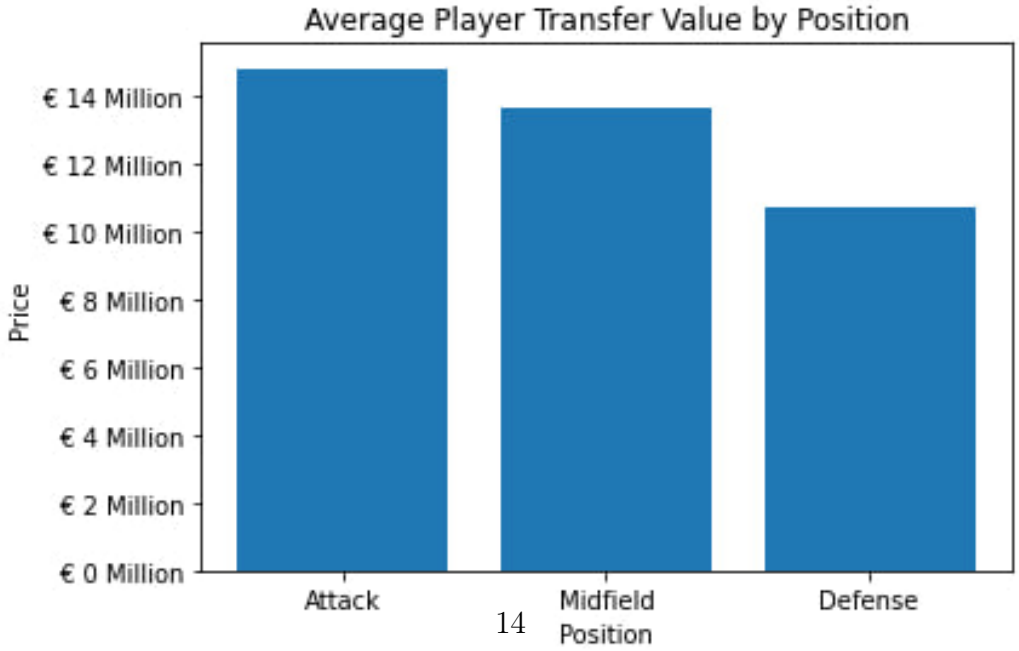
4.1 Wizualizacja i analiza danych pobranych z Transfermarktu



Rysunek 1: Liczba piłkarzy na każdej pozycji

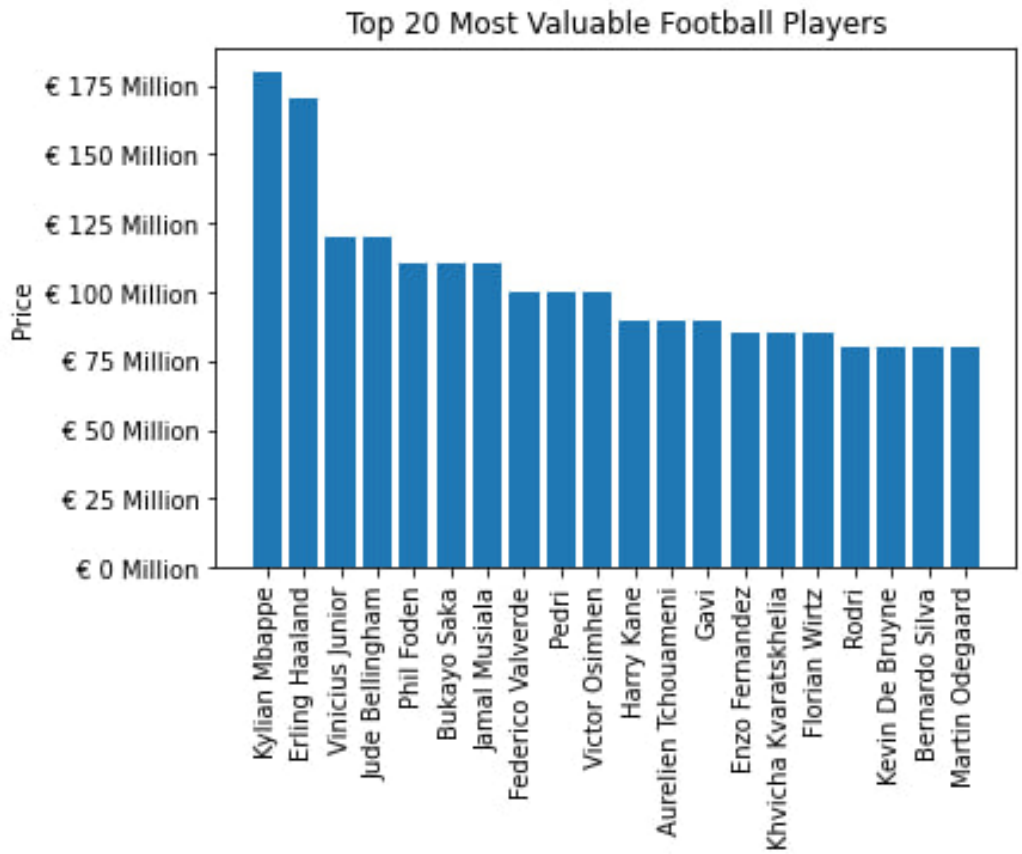


Rysunek 2: Liczba piłkarzy z podziałem na podstawowe pozycji

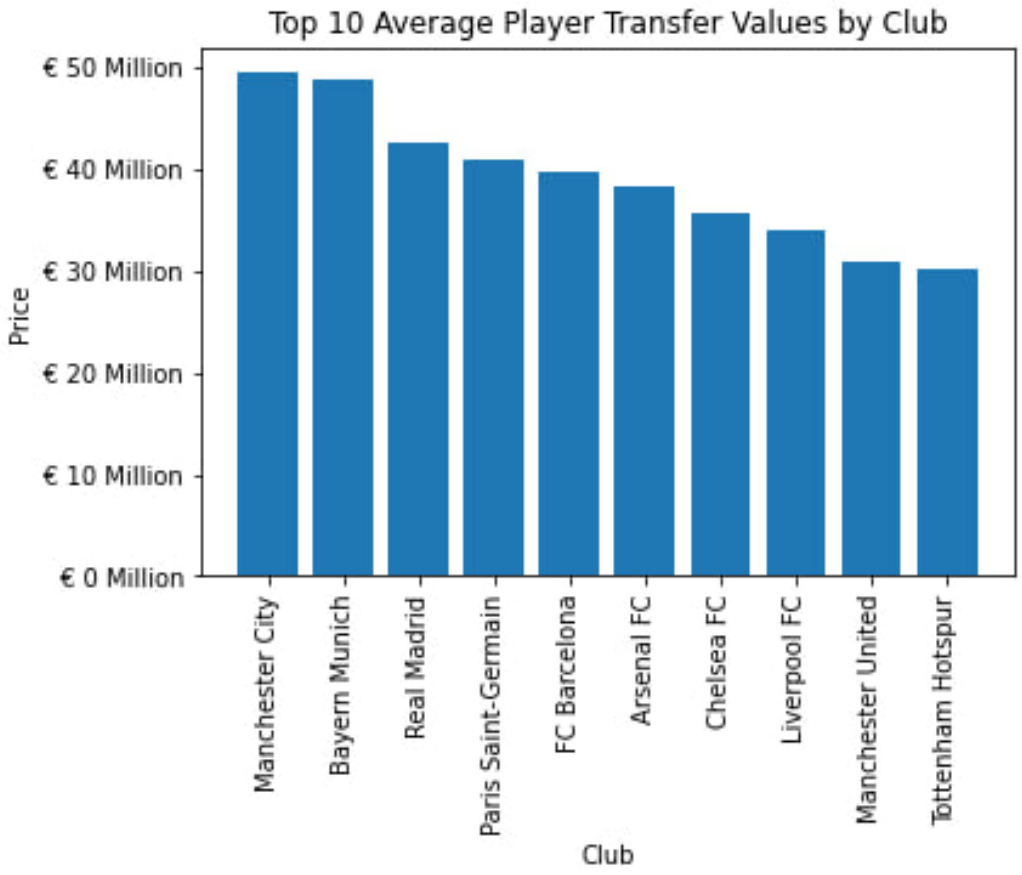


Rysunek 3: Średnia wartość transferu zawodnika według pozycji

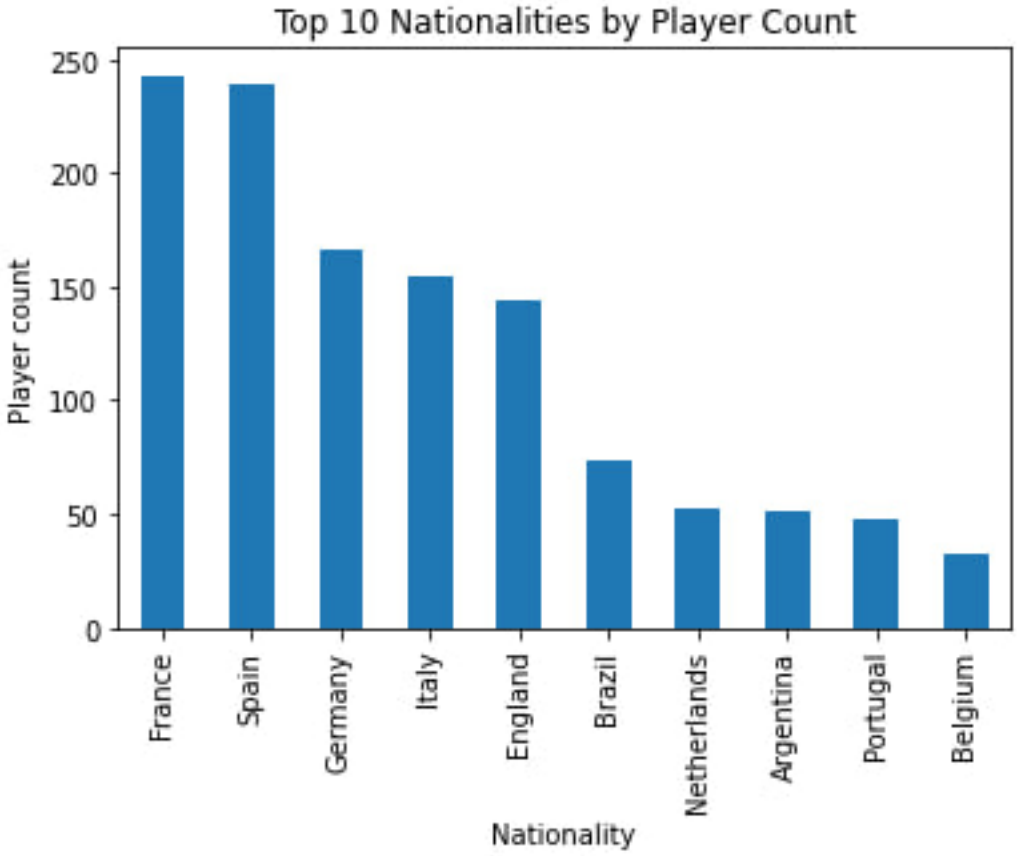
Patrząc na wykresy 2 oraz 3 można zobaczyć, że napastników jest mniej niż innych piłkarzy, ale są oni najdrożsi, i naodwrót: obrońców jest więcej niż innych piłkarzy, ale są oni najtansi.



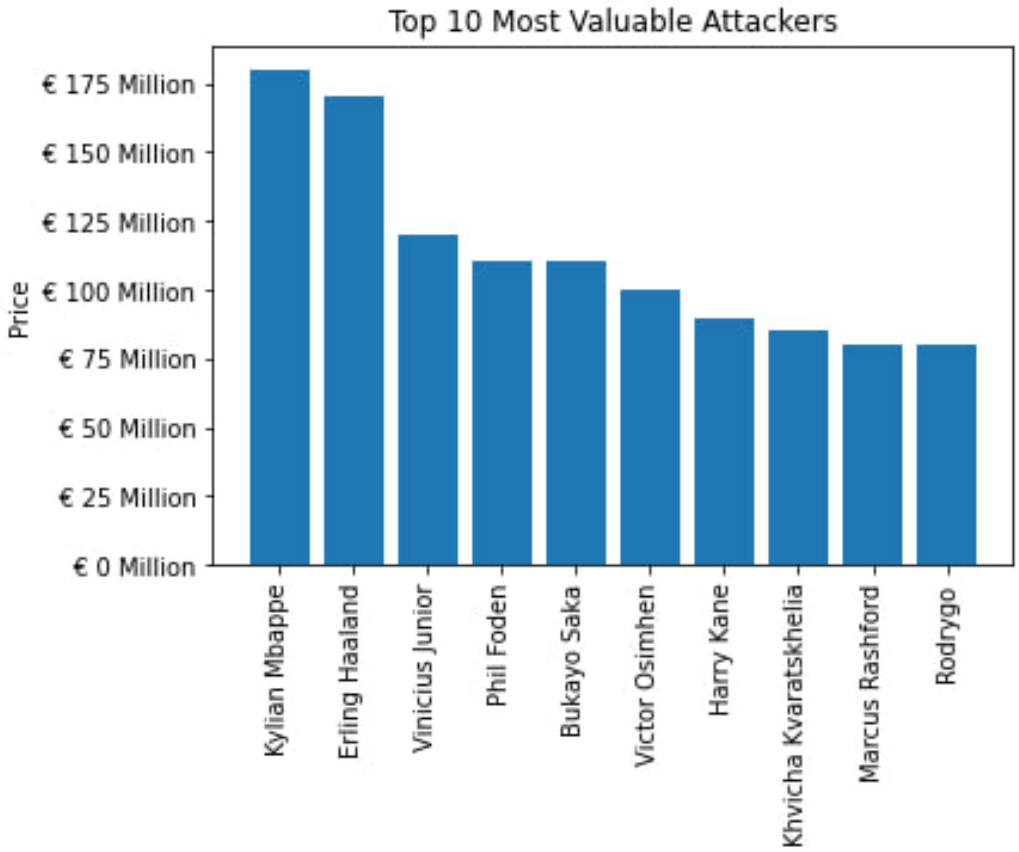
Rysunek 4: 20 najbardziej wartościowych piłkarzy



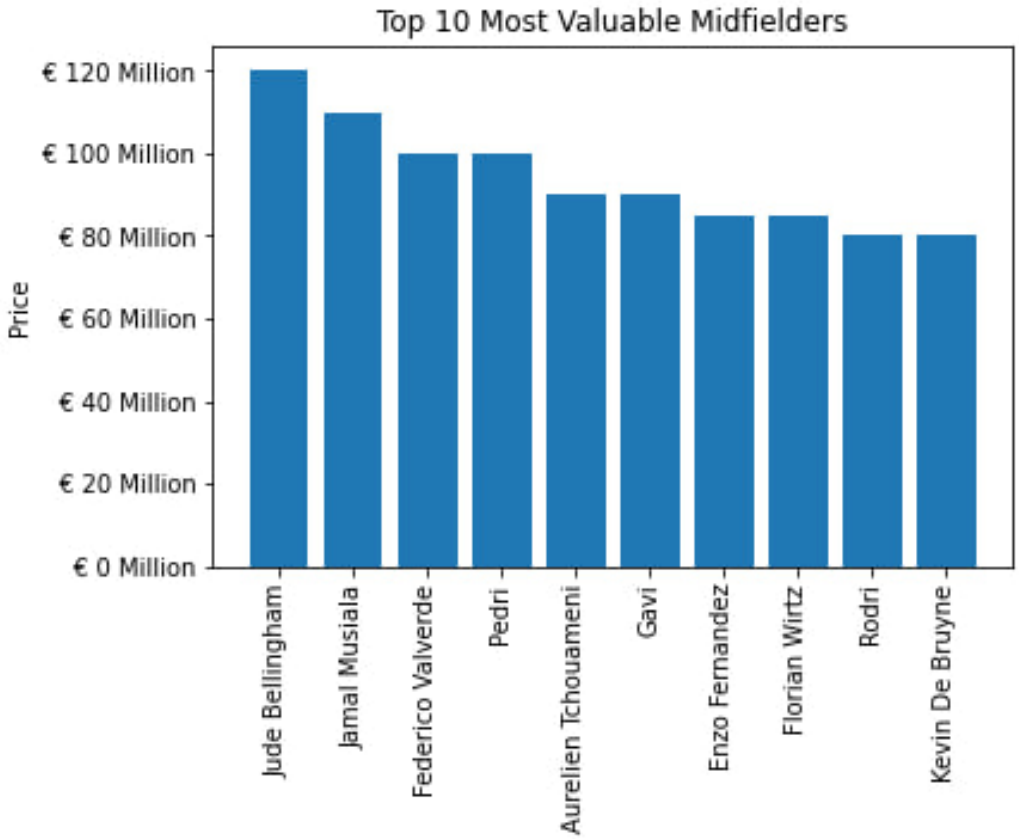
Rysunek 5: 10 najlepszych średnich wartości transferowych zawodników według klubów



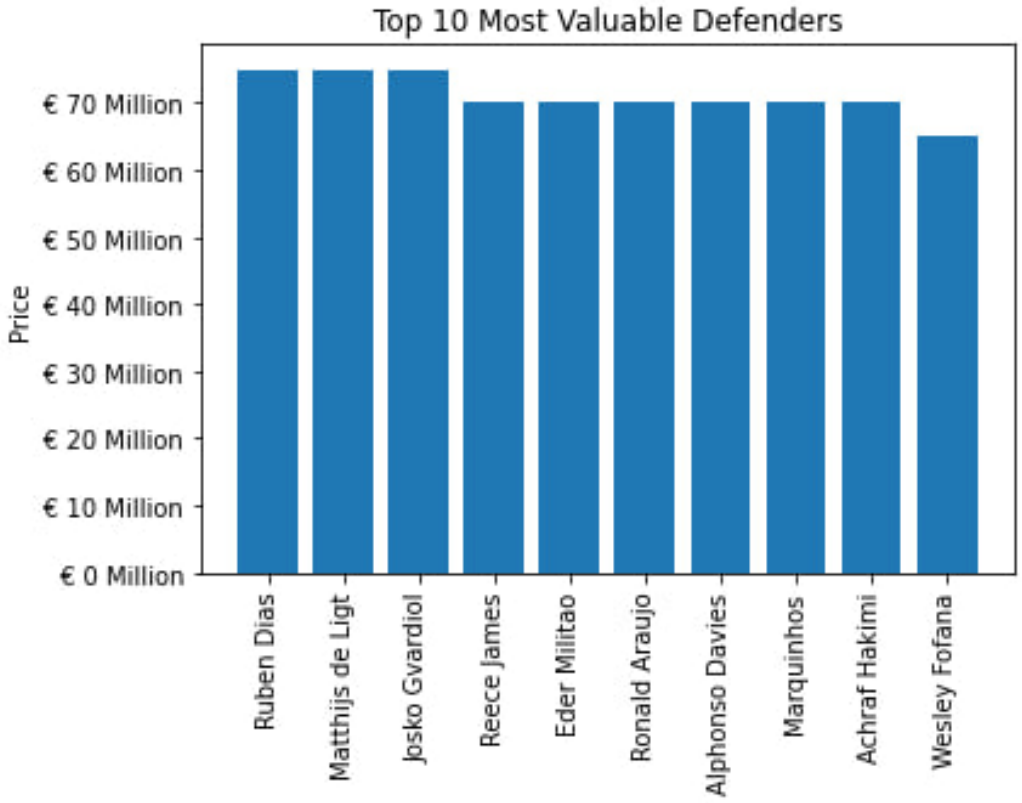
Rysunek 6: 10 najważniejszych narodowości według liczby graczy



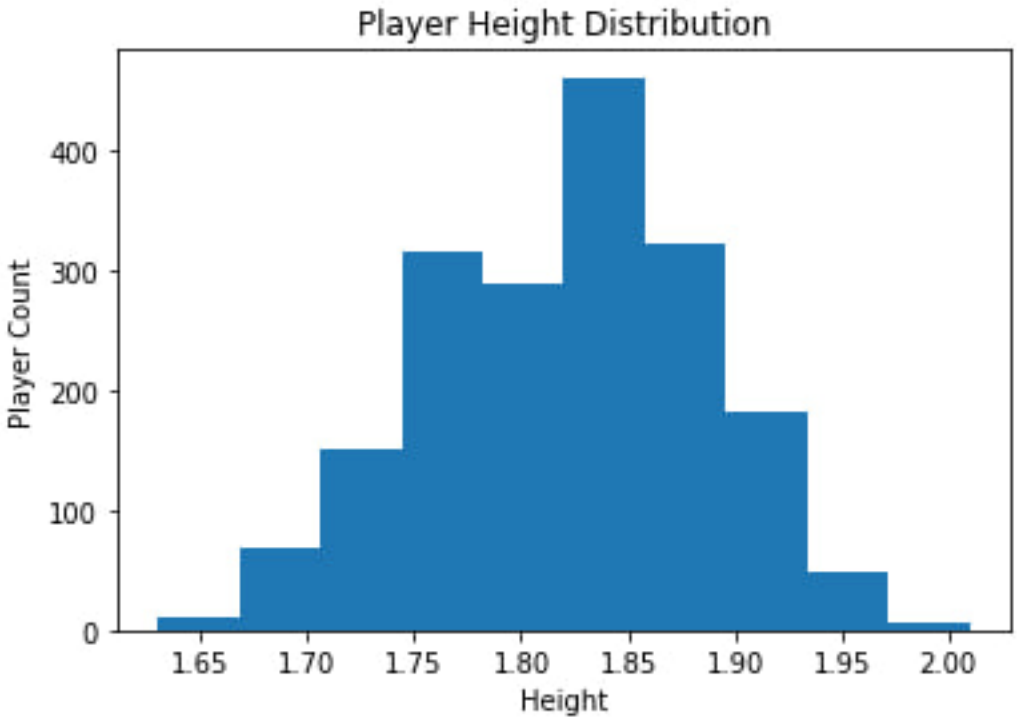
Rysunek 7: 10 najbardziej wartościowych napastników



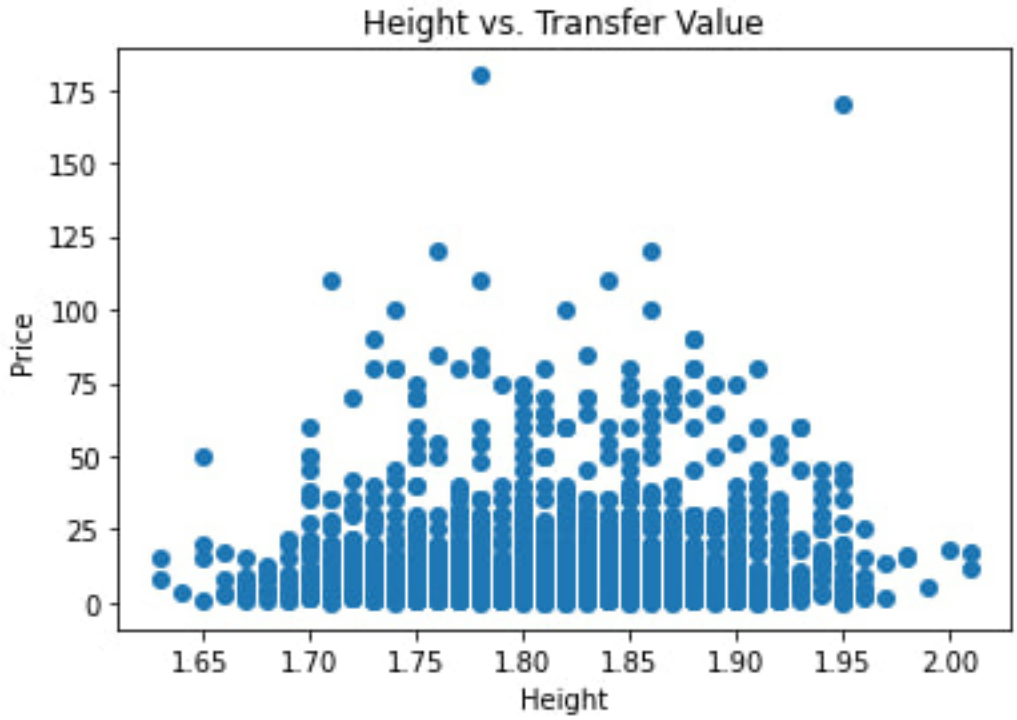
Rysunek 8: 10 najbardziej wartościowych pomocników



Rysunek 9: 10 najbardziej wartościowych obrońców

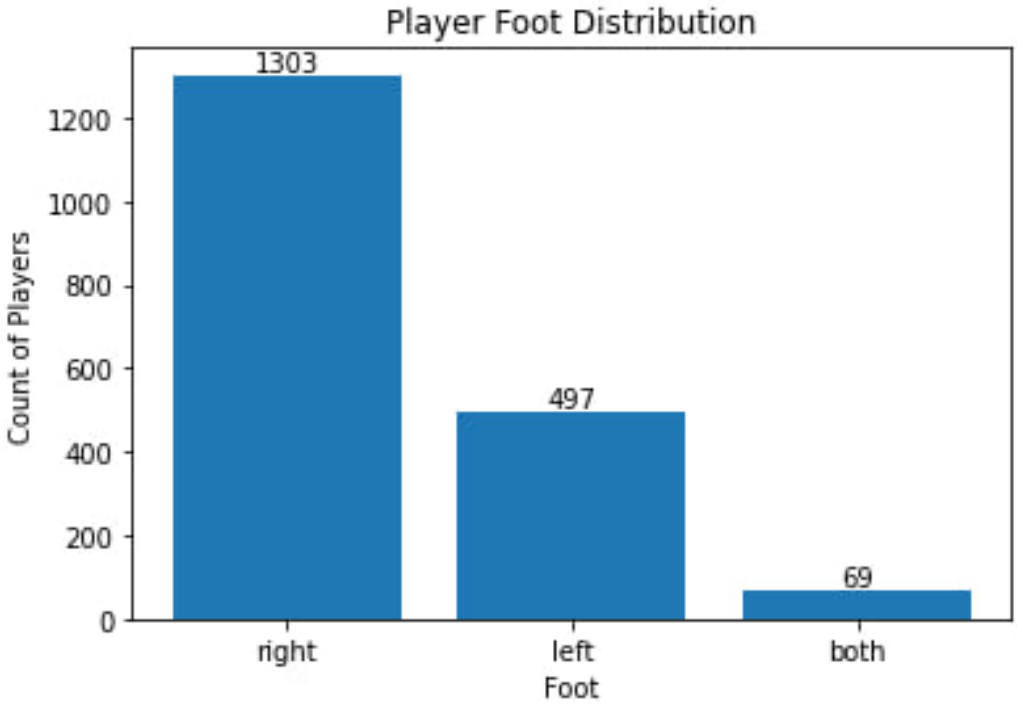


Rysunek 10: Rozkład wzrostu zawodników

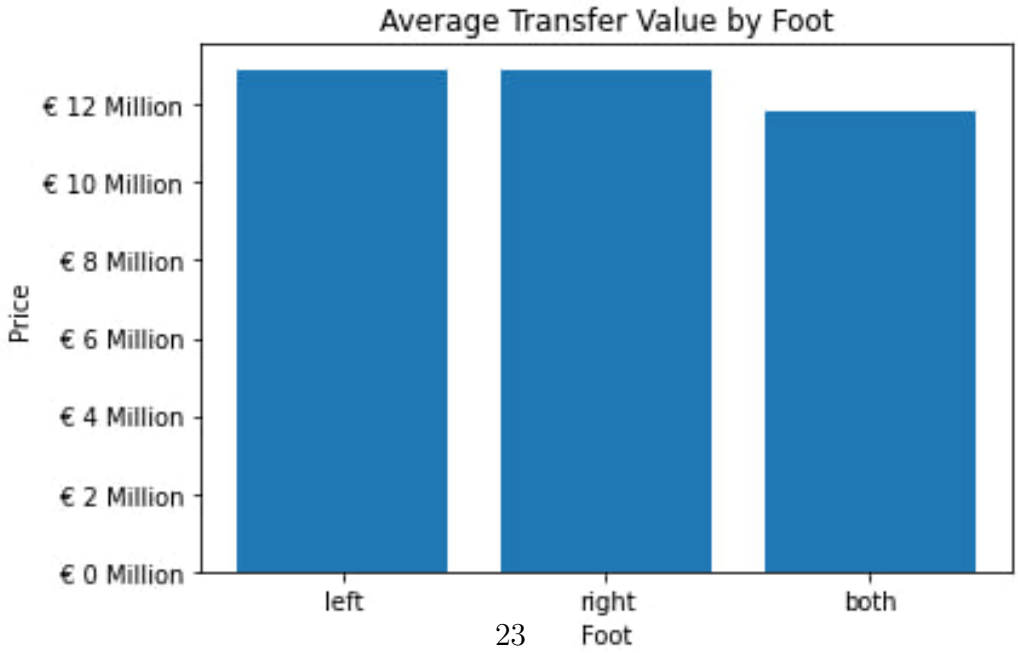


Rysunek 11: Wysokość a wartość transferu

Na wykresie 11 można zobaczyć, że w środku wykresu istnieją wartości większe od innych. Jest to prawdopodobnie dlatego że piłkarzy ze wzrostem około 1.85 m jest najwięcej, jak można zobaczyć na wykresie 10, i jest to logicznie, że im więcej jest piłkarzy, tym więcej jest drogich piłkarzy.

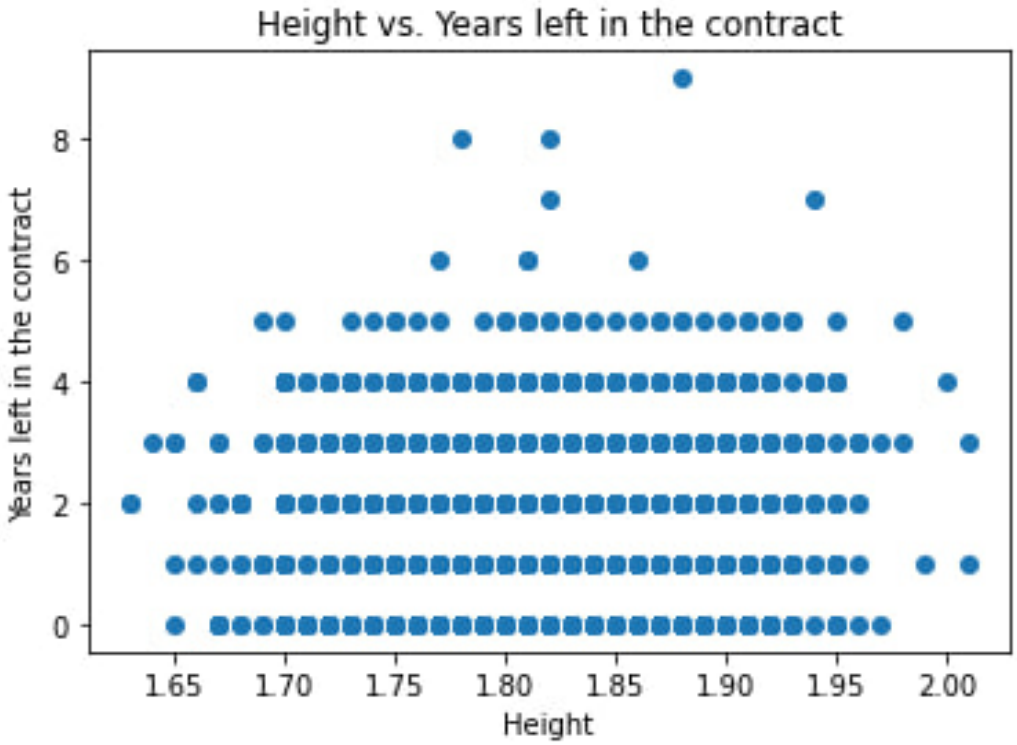


Rysunek 12: Rozkład głównych stóp graczy



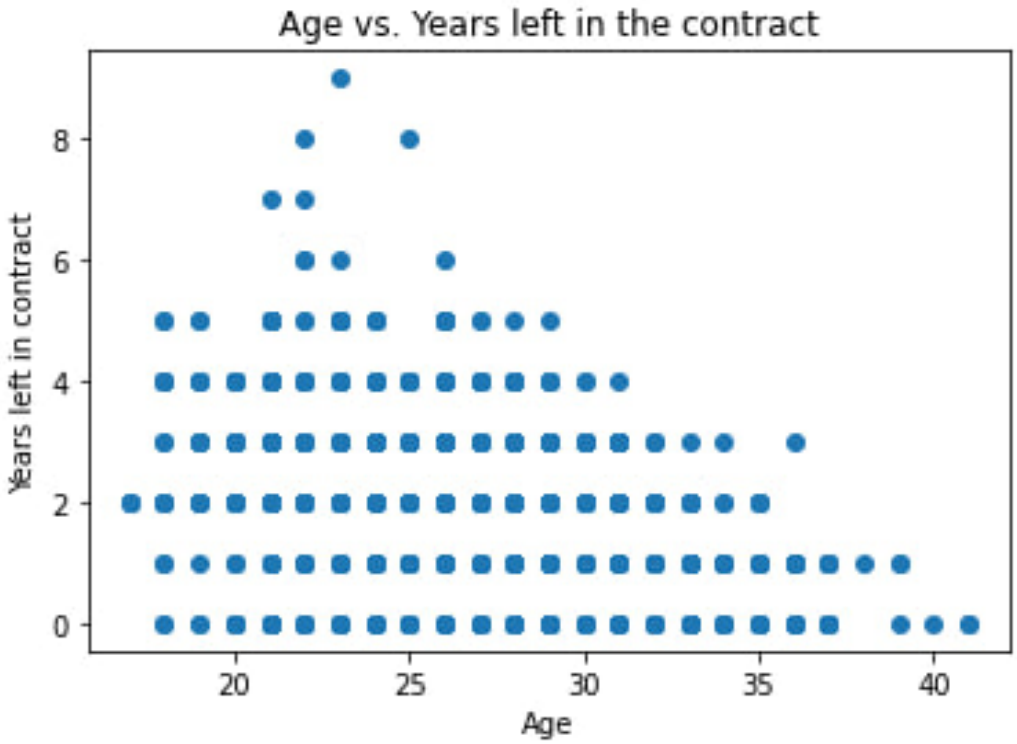
Rysunek 13: Średnia wartość transferu według głównej stopy

Na wykresie 13 można zobaczyć, że wartość piłkarza nie zależy od jego głównej stopy



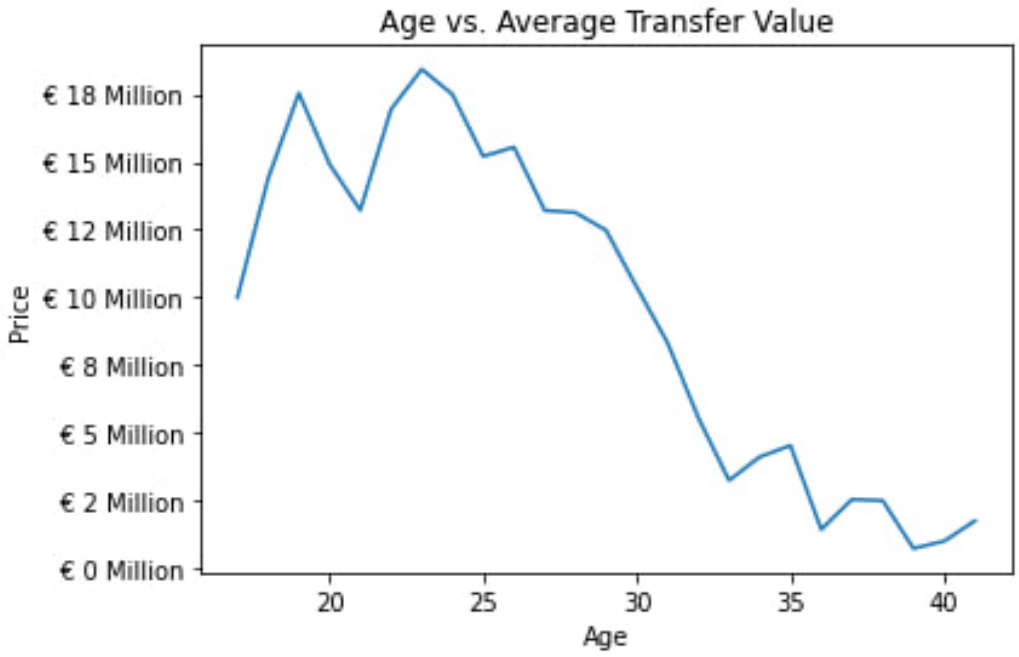
Rysunek 14: Wysokość a czas do wygaśnięcia kontraktu

Na wykresie 14 można zobaczyć, że czas do wygaśnięcia kontraktu piłkarza nie zależy od jego wzrostu



Rysunek 15: Wiek a czas do wygaśnięcia kontraktu

Na wykresie 15 można zobaczyć, że im młodszy jest piłkarz, tym więcej czasu piłkarz ma do wygaśnięcia kontraktu i naodwrót, im starszy jest piłkarz, tym mniej czasu pozostaje mu do wygaśnięcia kontraktu. Jest to logicznie, ponieważ kluby chcą mieć młodszych piłkarzy jak dłużej u siebie, dlatego mają długie kontrakty, a w przypadku z piłkarzami którzy

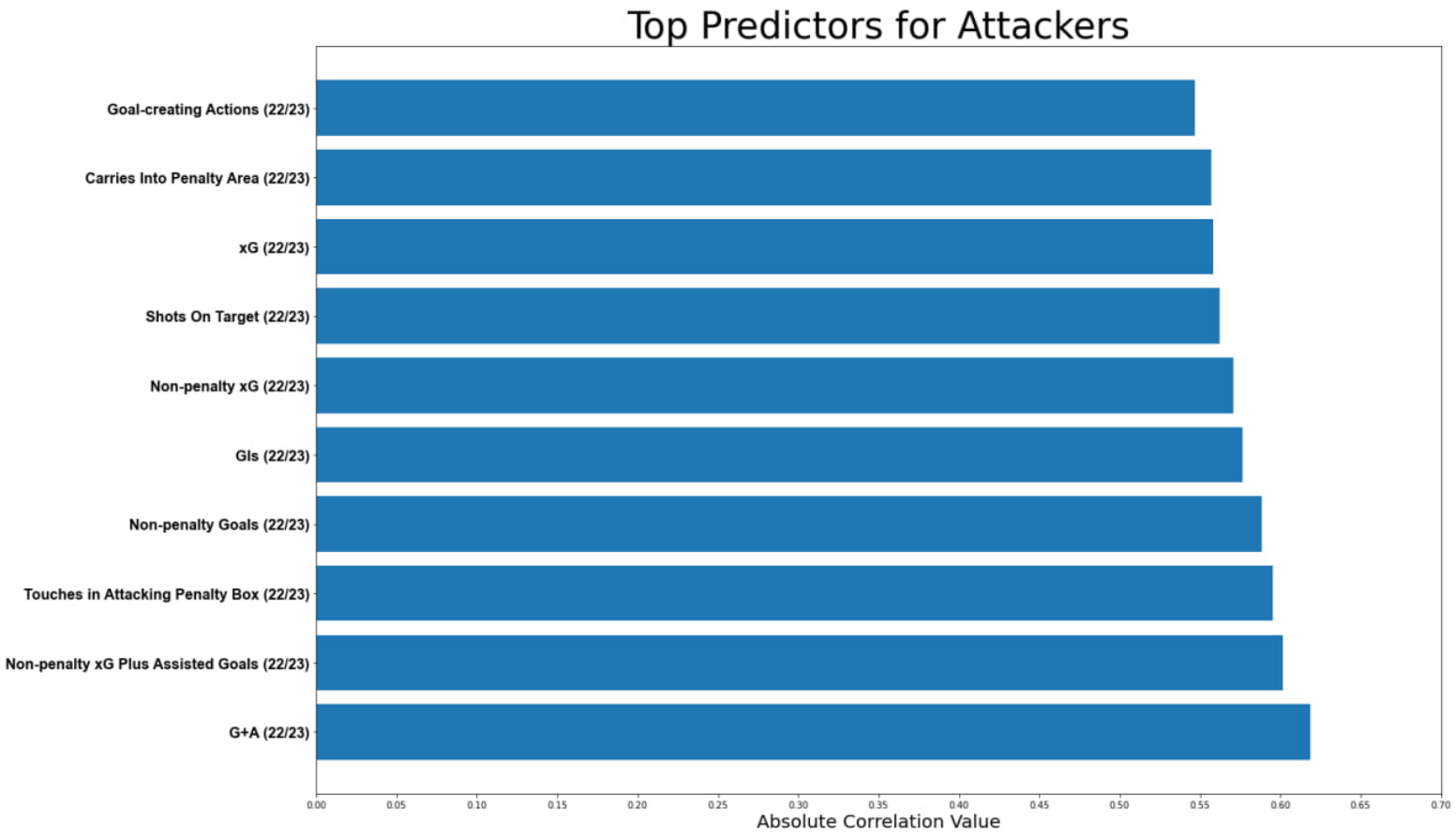


Rysunek 16: Wiek a średnia wartość transferu

Rysunek 17: Czas do wygaśnięcia kontraktu a średnia wartość transferu

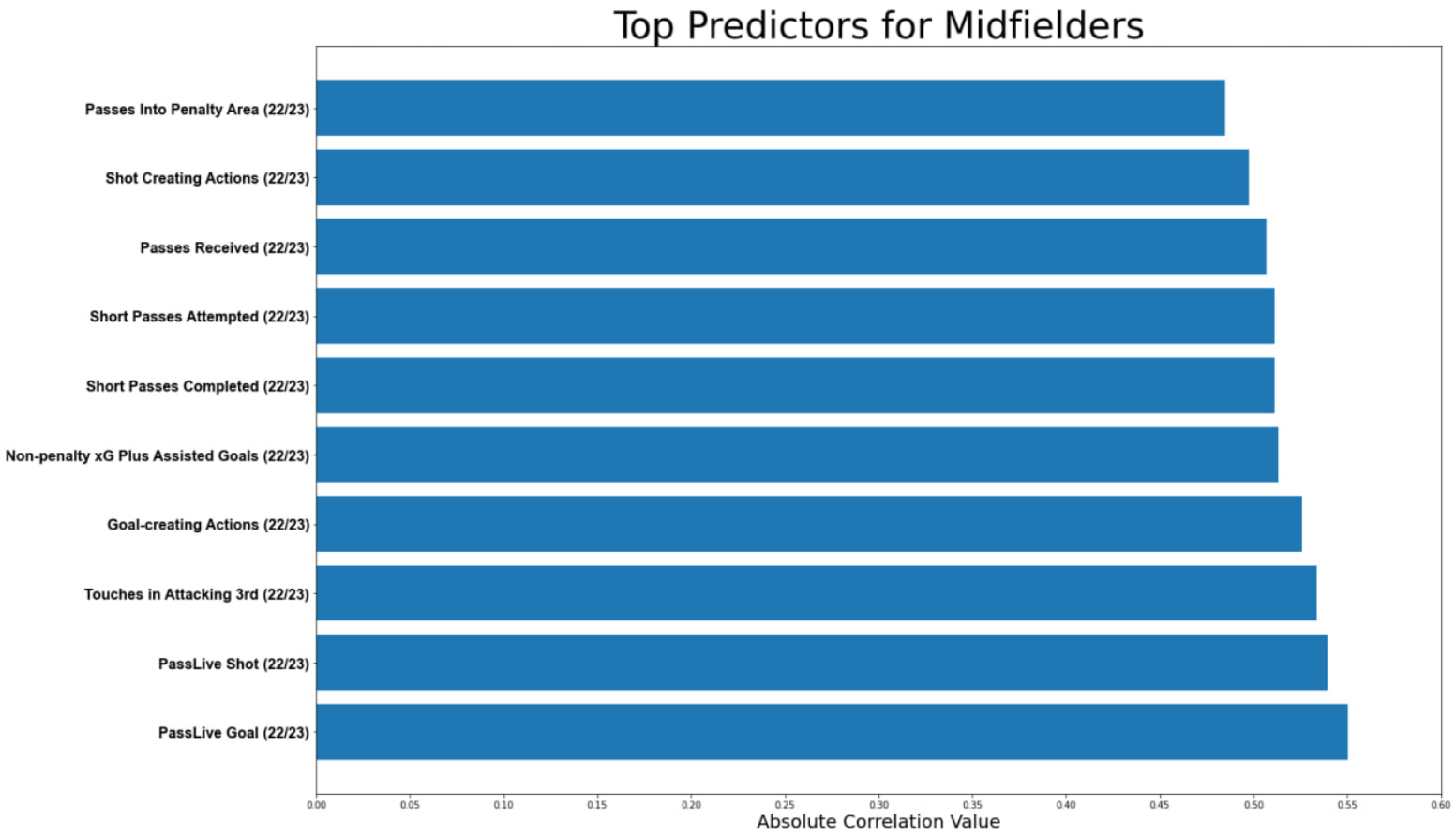
4.2 Wizualizacja i analiza danych pobranych z FBREF

Poniżej są wykresy, które pokazują najlepsze cechy dla piłkarzy (czyli te cechy, które najbardziej korelują z wartością transferową) w zależności od ich pozycji na boisku. Brałem pod uwagę tylko ostatni sezon, ponieważ chcę mieć top unikalnych cech.



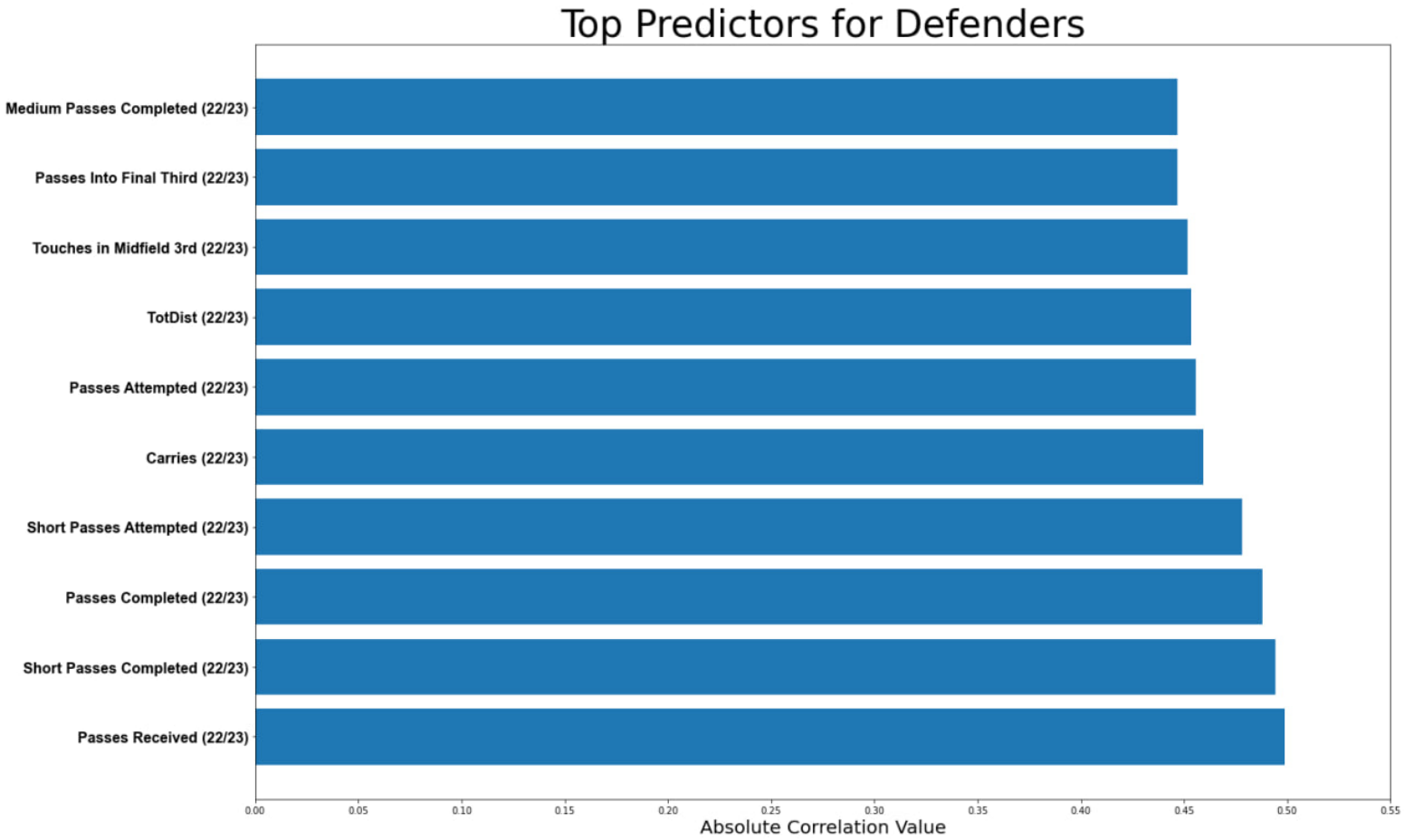
Rysunek 18: Najlepsze cechy dla napastników

Na wykresie 18 widać, że cecha, która najbardziej wpływa na wartość transferową napastnika, to suma goli i asystów, co nie jest zaskoczeniem.



Rysunek 19: Najlepsze cechy dla pomocników

Na wykresie 19 widać, że dla pomocnika najlepsze cecha to PassLive Goal, czyli ukończone podania prowadzące do piłki. Nie jest to też zaskoczeniem, bo pomocnicy są oceniani na podstawie ich zdolności do dyktowania tempa gry i pozytywnego ułatwiania drużynie postępów w rozgrywaniu piłki.



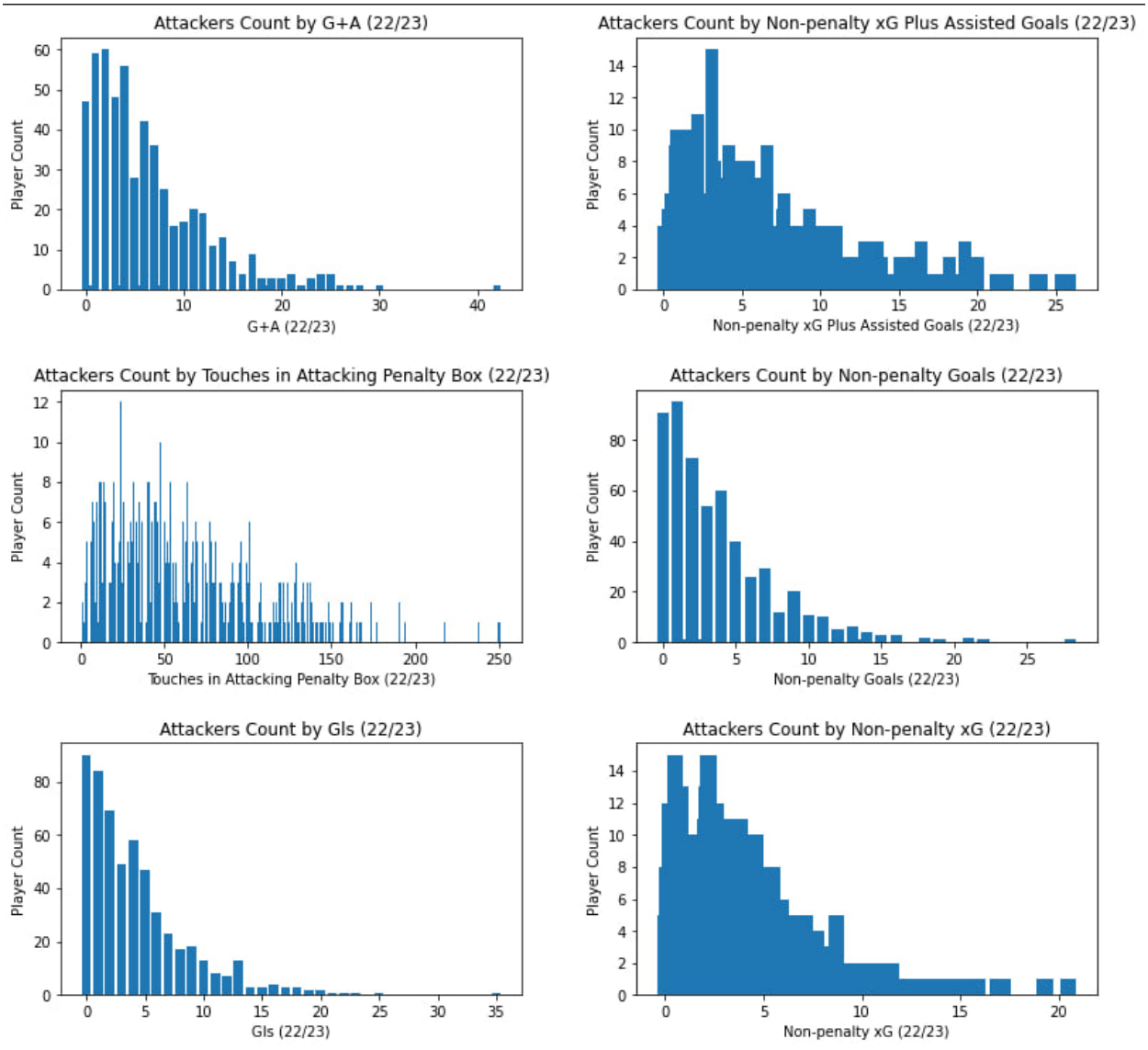
Rysunek 20: Najlepsze cechy dla obrońców

Na wykresie 20 widać, że najlepsza cecha dla obrońcy to otrzymane podania i ten fakt był trochę niesamowity dla mnie, bo myślałem, że będzie to dryblerzy pokonani (dribblers tackled) . Myślę, że wynika to z tego faktu, że najdrożsi piłkarzy występują za najlepsze kluby, które bardzo dużo prowadzą piłkę, i większą część meczu zamiast bronić się, obrońcy przeprowadzają ataki, dlatego mają dużo otrzymanych podań oraz podań do innych piłkarzy.

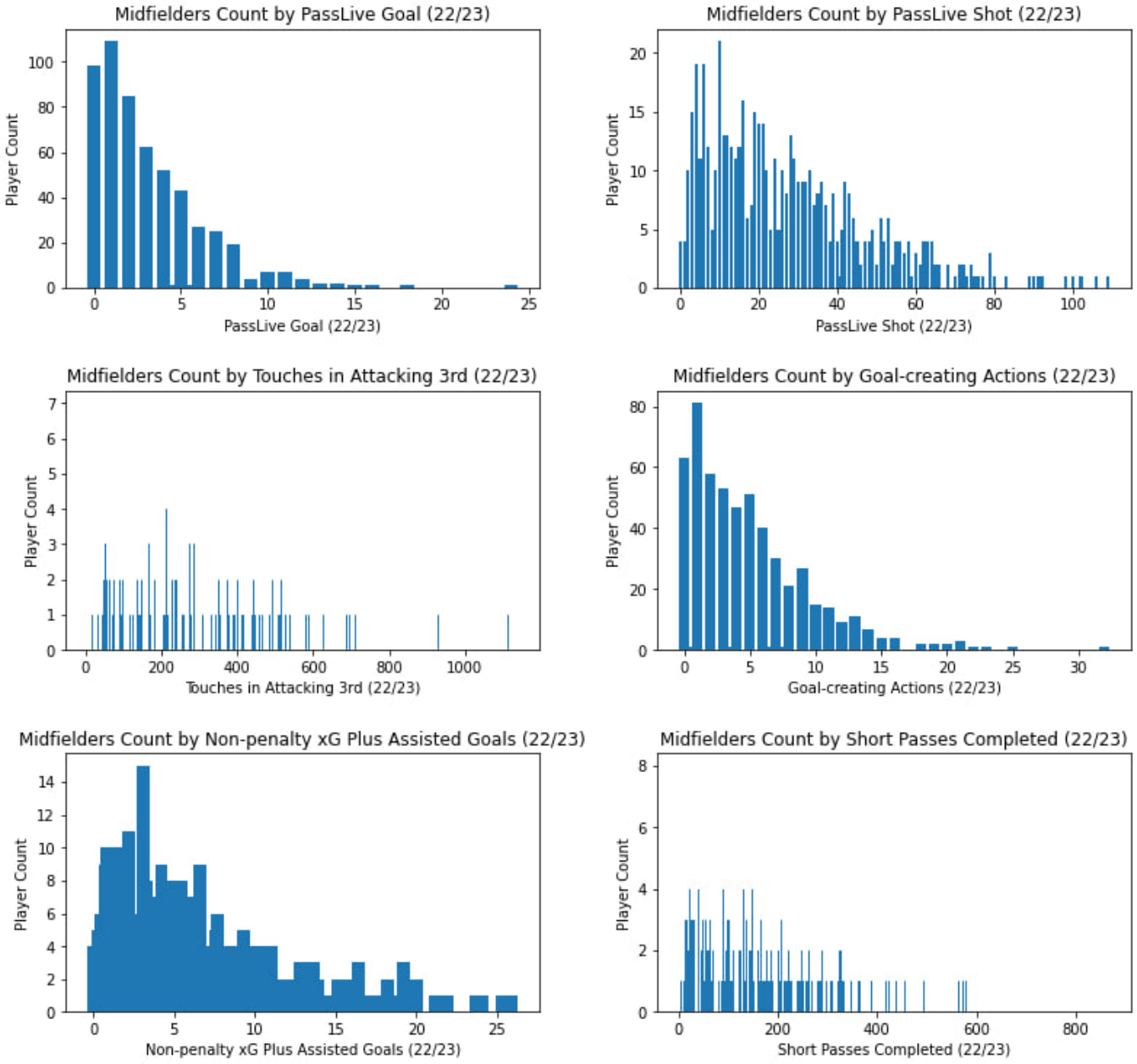
5 Pre-processing oraz przygotowanie modeli

5.1 Pre-processing danych

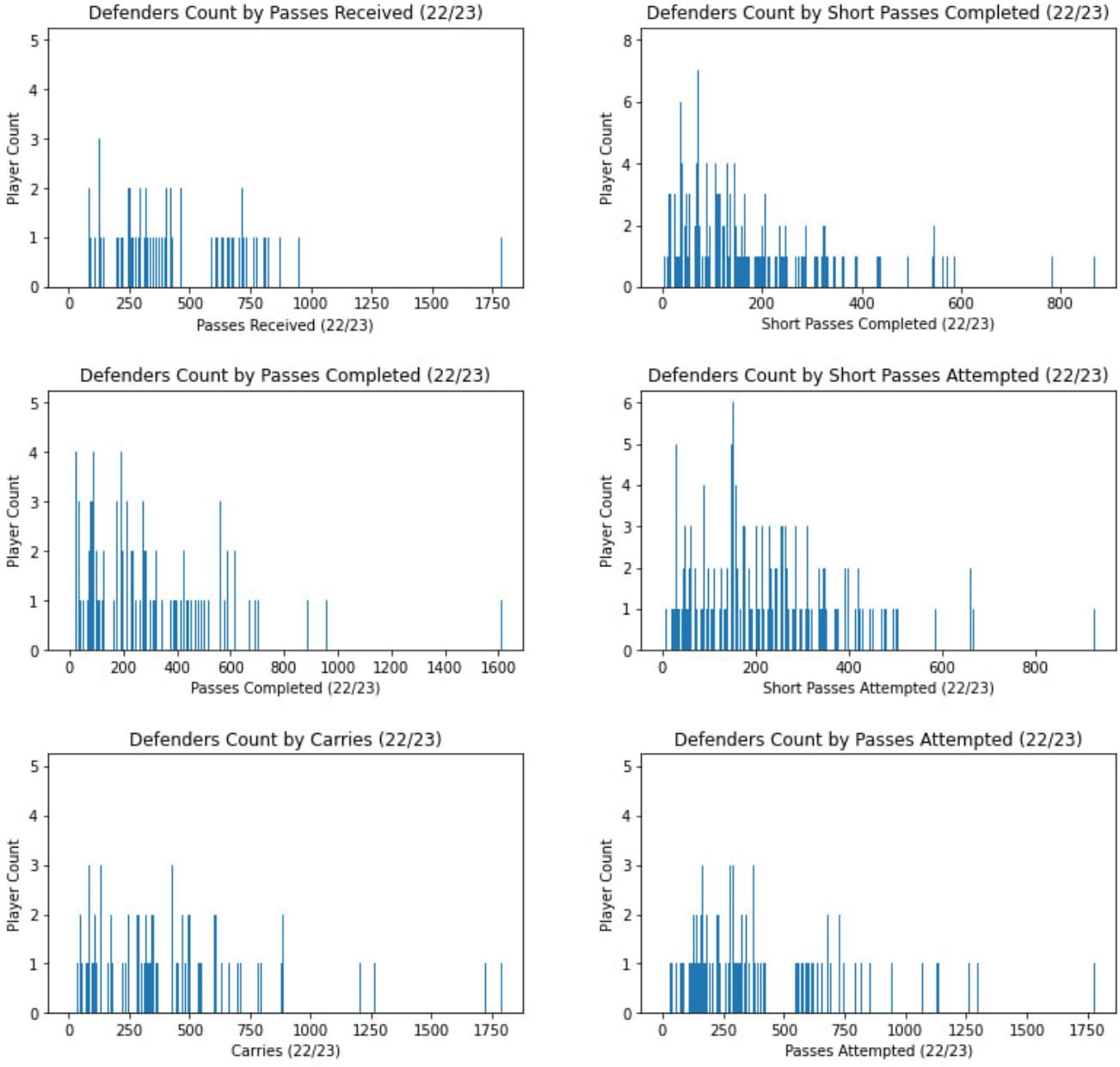
Poniżej są wykresy, pokazujące dystrybucję danych dla najlepszych cech



Rysunek 21: Dystrybucja danych dla najlepszych cech napastników



Rysunek 22: Dystrybucja danych dla najlepszych cech pomocników



Rysunek 23: Dystrybucja danych dla najlepszych cech obrońców

Na wykresach 21, 22 oraz 23 można zobaczyć, że cechy, które mają największy wpływ na wartość transferową piłkarzy, mają znaczące odchylenie w

rozkładzie wartości. Pokazuje to, że cechy musiałyby zostać przekształcone, aby uzyskać rozkład normalny bardziej podobny do rozkładu Gaussa, zanim będzie można je wykorzystać do dopasowania i trenowania modeli.

Aby osiągnąć pożądany rozkład normalny podobny do rozkładu Gaussa, użyto PowerTransformer z biblioteki wstępnego przetwarzania sklearn. Jest to narzędzie, które przekształca dane tak, aby były bardziej podobne do rozkładu gaussowskiego. Jest to przydatne do rozwiązywania problemów związanych z modelowaniem danych, gdzie wariancja danych w kolumnie predyktora nie jest stała (jak w moim przypadku)

Po uczynieniu rozkładu normalnego w cechach, użyłem Robust Scaler do standaryzacji danych. Robust Scaler jest szczególnie przydatny w przypadku cech, w których występują wartości odstające, które mogą źle skutkować na predykcję. Usuwa on medianę, a następnie skaluje dane kolumnowe zgodnie z zakresem kwantyli.

5.2 Przygotowanie modeli

Do przewidywania wartości użyłem 7 różnych modeli z bibloteki sklearn w celu znalezienia modelu, który najlepiej przewiduje wartość transferową piłkarzy: Regresja liniowa, regresja Lasso, regresja grzbietowa, AdaBoost, GradientBoost, RandomForest oraz DeisionTree.

Zdecydowałem użyć każdego modelu do przewidywania wartości transferowych dla datasetu podzielonego na pozycję podstawowe (napastnicy, pomocnicy oraz obrońcy), wszystkie pozycji (ogólnie 12) oraz datasetu, który nie jest podzielony na pozycji, żeby zobaczyć, jak dobrze modele przewidują wartość transferową w zależności od posiadania lub nie posiadania informacji o pozycji piłkarza.

Wybrałem negatywny błąd bezwzględny (NMAE) jako metrykę obliczania błędu modelu. MAE mierzy medianę wartości bezwzględnych różnicy między przewidywanymi a rzeczywistymi wartościami:

$$\text{MAE} = \text{median}(|y_{\text{pred}} - y_{\text{true}}|) \tag{1}$$

NMAE jest po prostu przeciwnością MAE, czyli wynik jest pomnożony przez -1. Często używa się tej metryki w celu maksymalizacji oceny jakości modelu,

ponieważ większa wartość -RMSE oznacza mniejszy błąd.

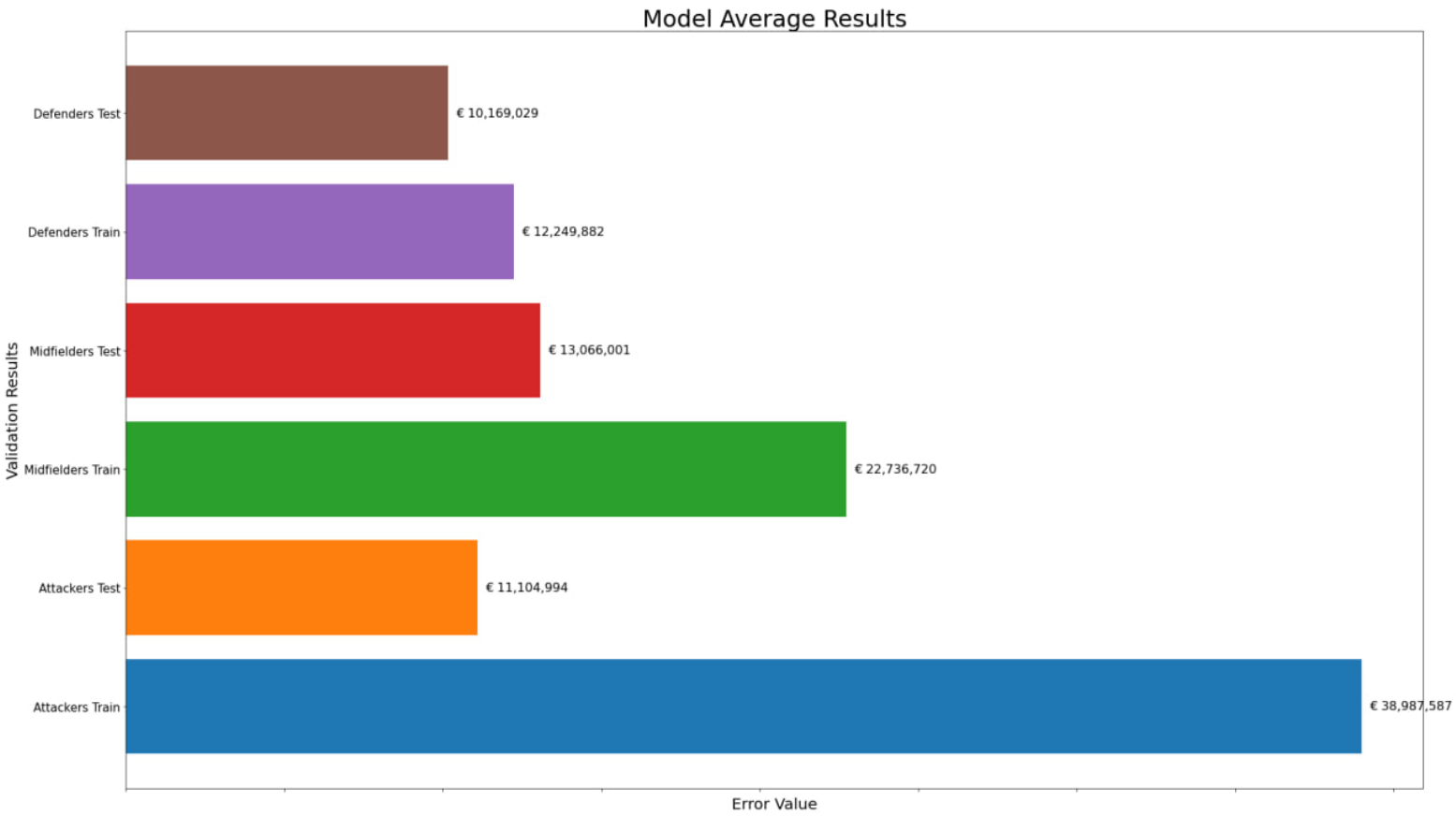
Wybrałem metrykę NMAE, ponieważ chcę znaleźć średnią różnicę pomiędzy wartością transferową przewidywalną a rzeczywistą.

Podzieliłem dane na treningowe i testowe ze współczynnikiem testSize 0.2, czyli dane testowe zawierają 1/5 danych datasetu, a dane treningowe - 4/5.

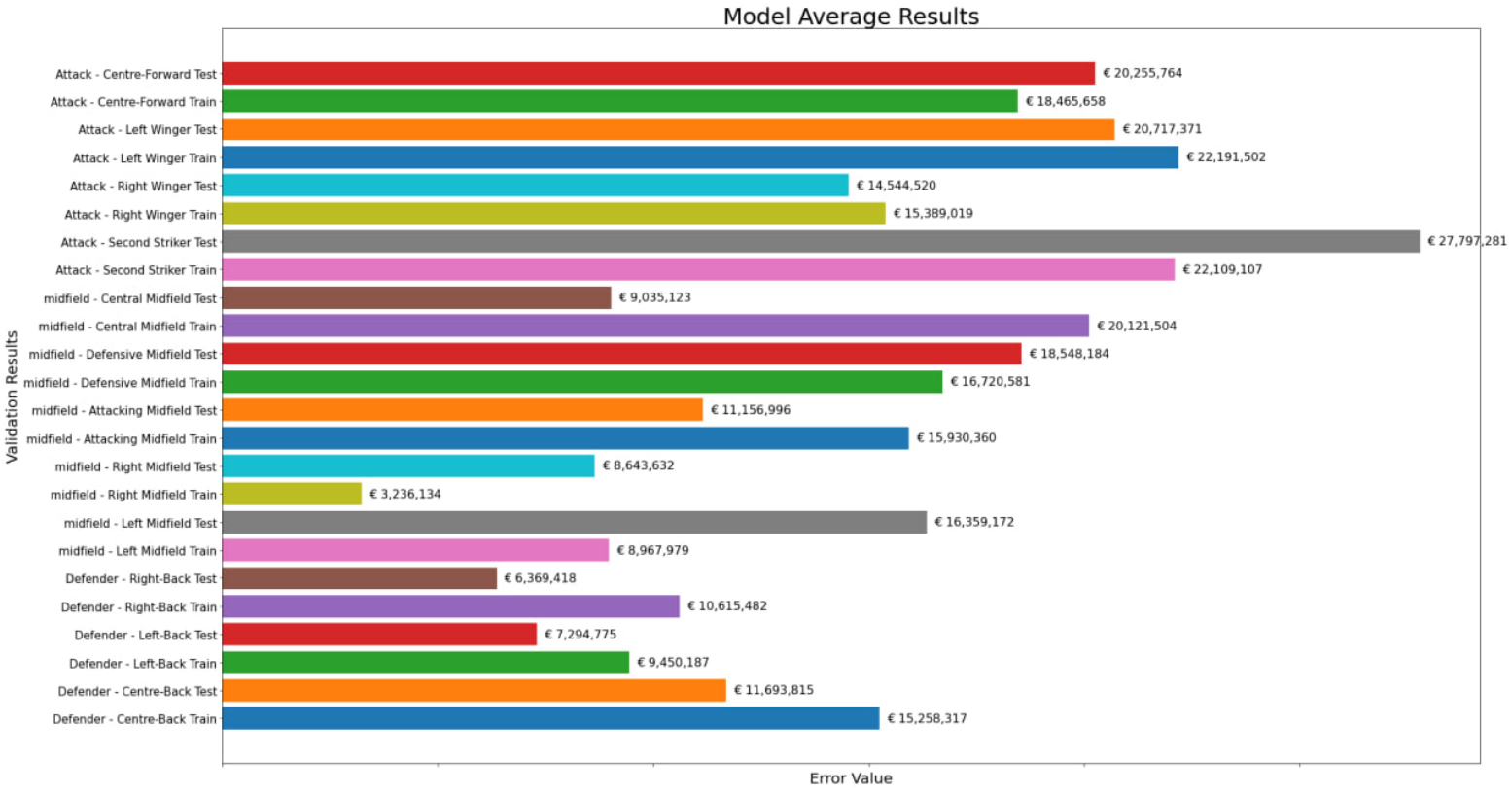
Do wyszukiwania NMAE użyłem funkcji crossValScore która wykonuje krzyżową walidację z parametrem cv=5, co znaczy, że dane są podzielone na 5 części i model jest trenowany i testowany na różnych kombinacjach tych podziałów. crossValScore zwraca tablicę zawierającą wyniki dla każdego podziału. Następnie, aby uzyskać średnią wartość błędu bezwzględnego dla zbioru treningowego i testowego, oblicza się średnia wartość wyników krzyżowej walidacji poprzez wywołanie mean() na tablicy. W końcu użyłem negacji, aby przekształcić wyniki w wartości dodatnie.

5.3 Rezultaty modeli

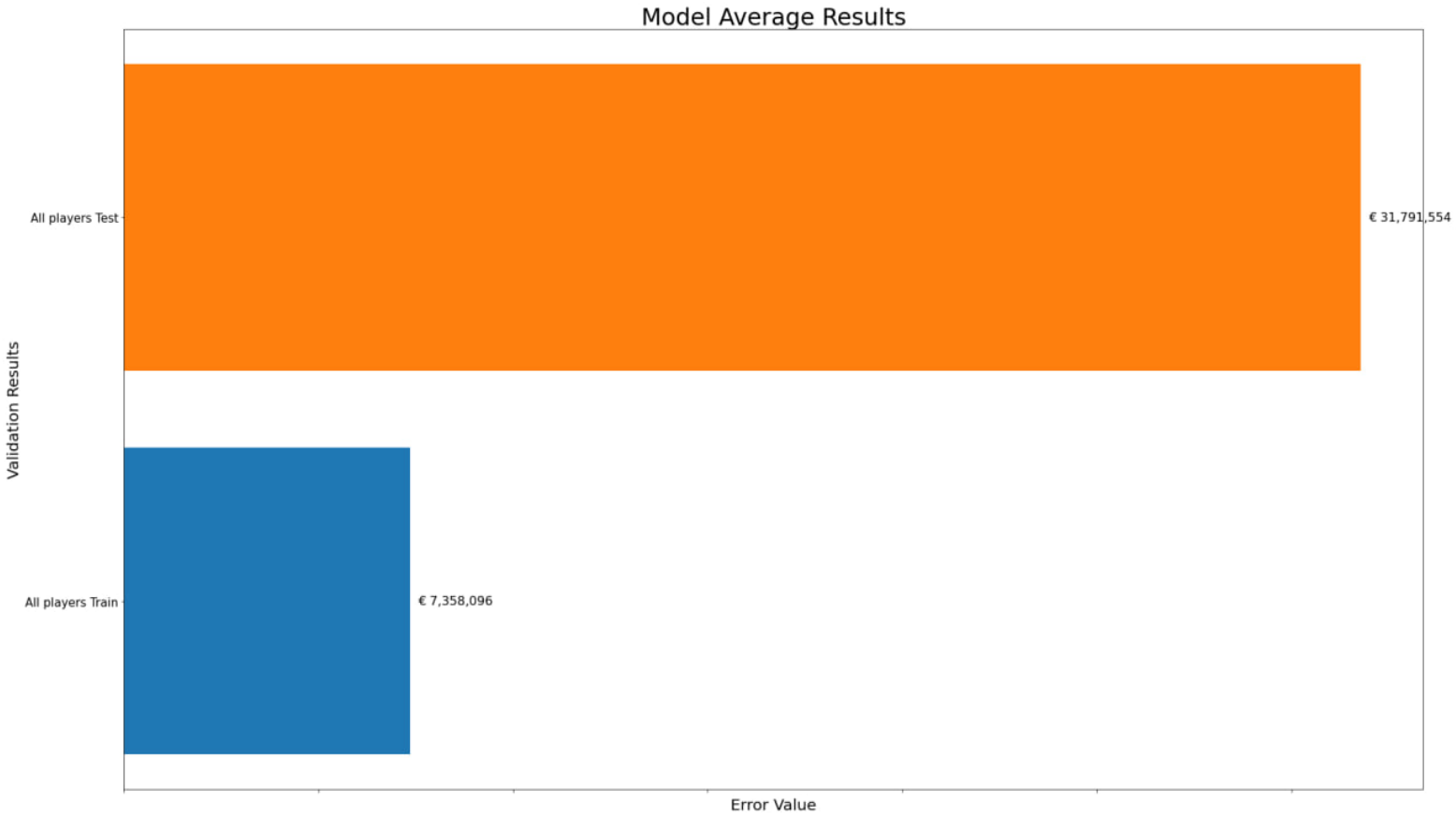
5.3.1 Rezultaty regresji liniowej



Rysunek 24: Błąd modeli wykorzystującej regresję liniową dla podstawowych pozycji

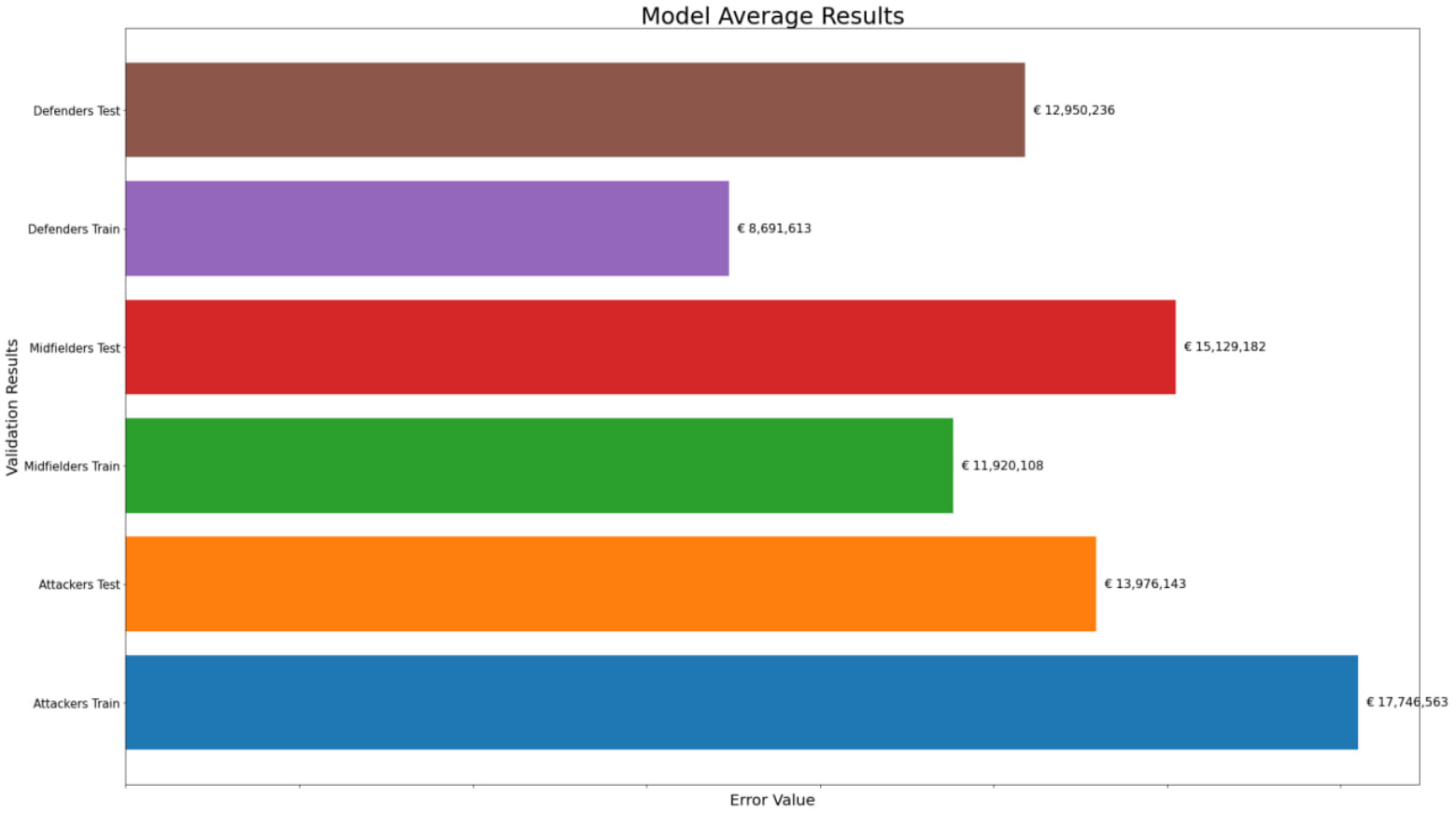


Rysunek 25: Błąd modeli wykorzystującej regresję liniową dla wszystkich pozycji

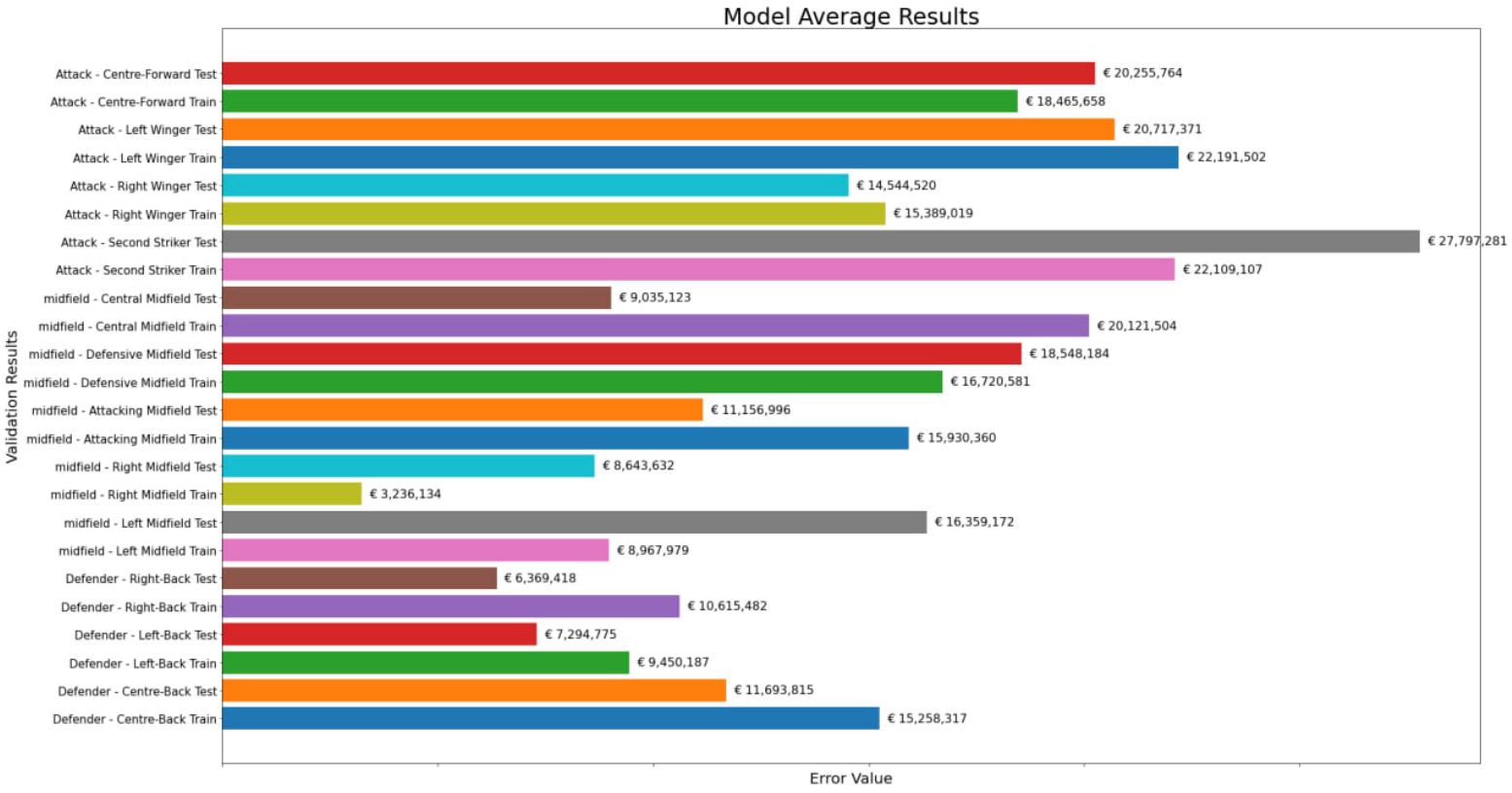


Rysunek 26: Błąd modeli wykorzystującej regresję liniową bez uwzględnienia pozycji

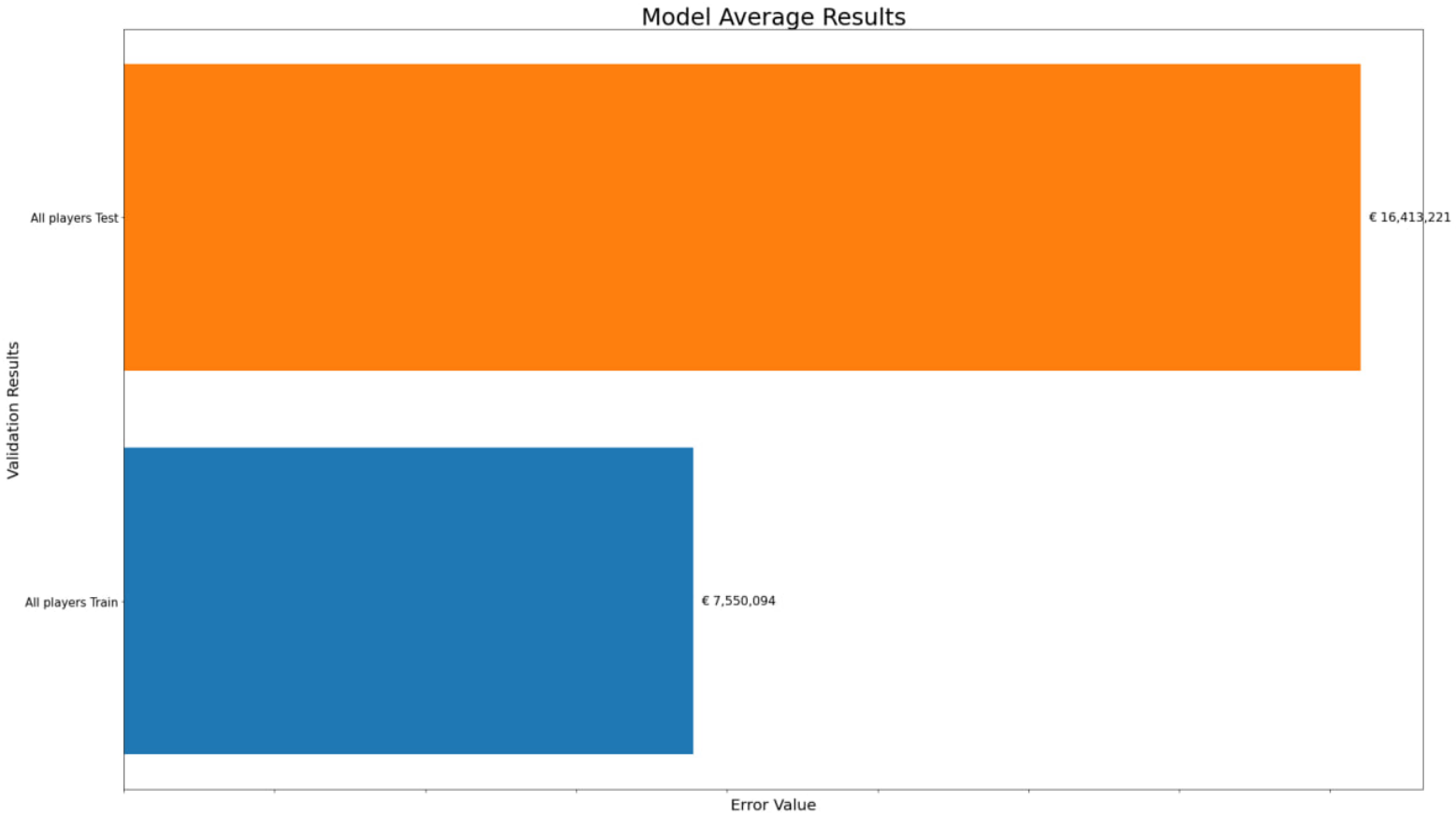
5.3.2 Rezultaty regresji Lasso



Rysunek 27: Błąd modeli wykorzystującej regresję Lasso dla podstawowych pozycji

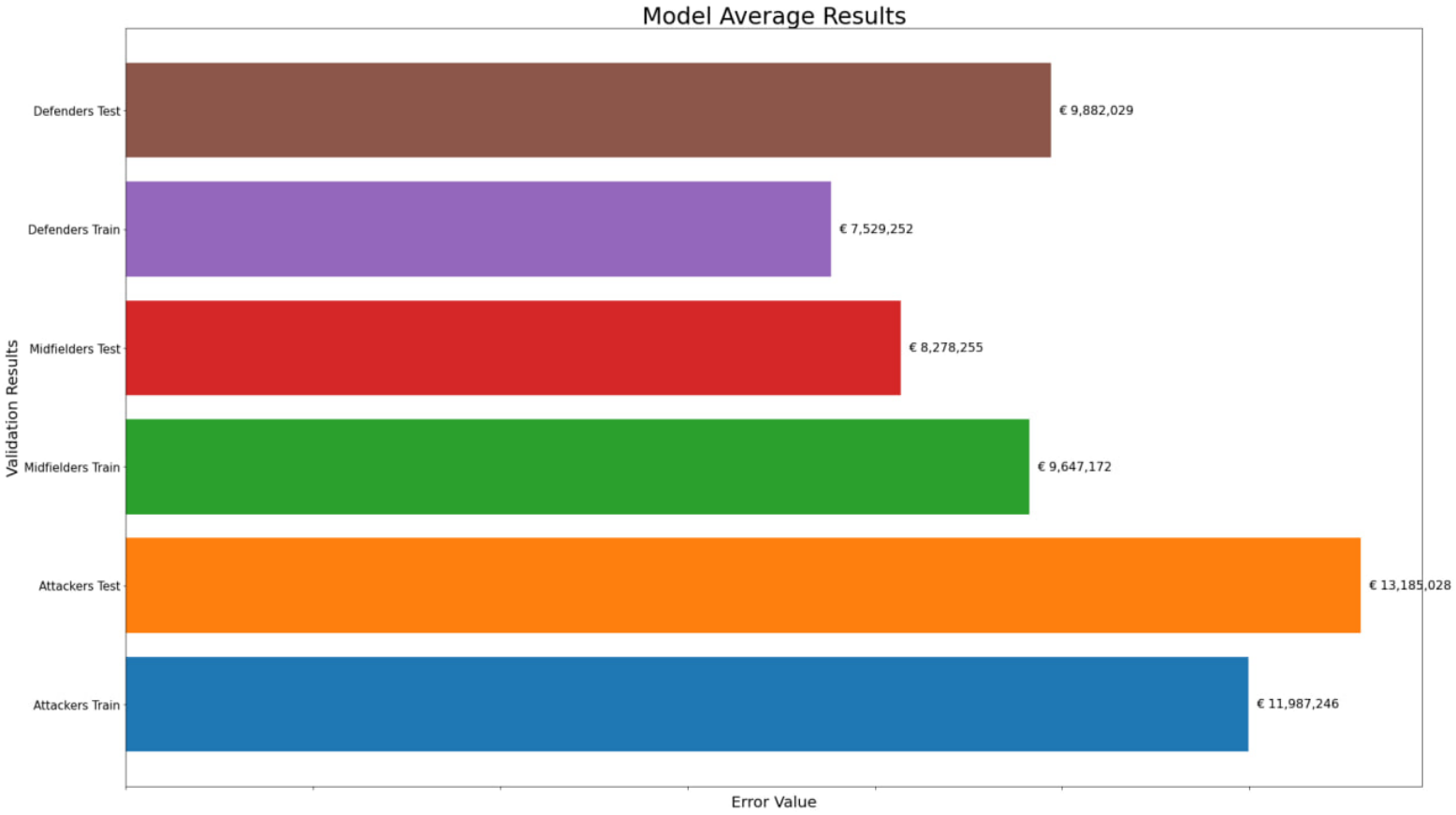


Rysunek 28: Błąd modeli wykorzystującej regresję Lasso dla wszystkich pozycji

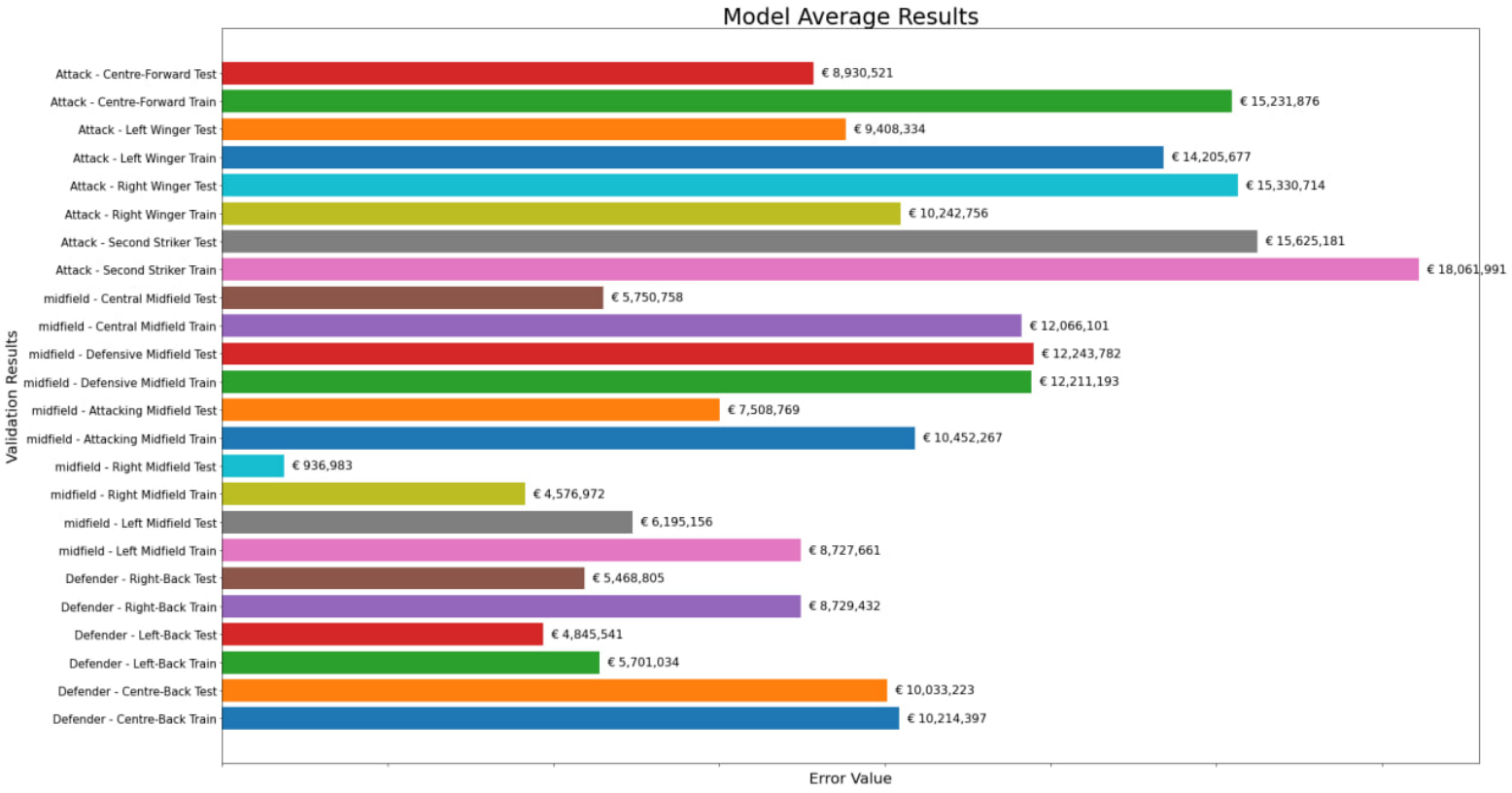


Rysunek 29: Błąd modeli wykorzystującej regresję Lasso bez uwzględnienia pozycji

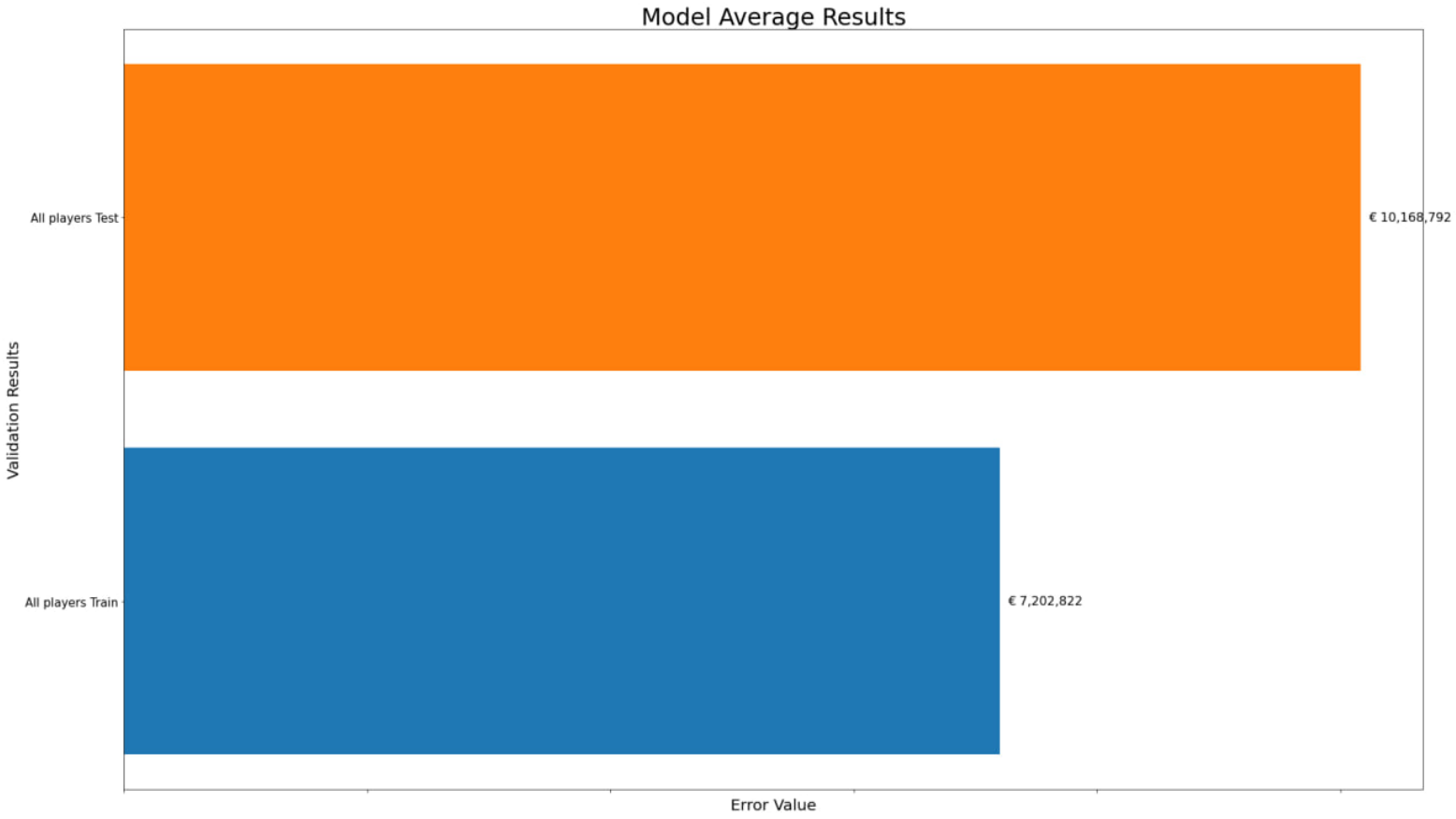
5.3.3 Rezultaty regresji grzbietowej



Rysunek 30: Błąd modeli wykorzystującej regresję grzbietową dla podstawowych pozycji

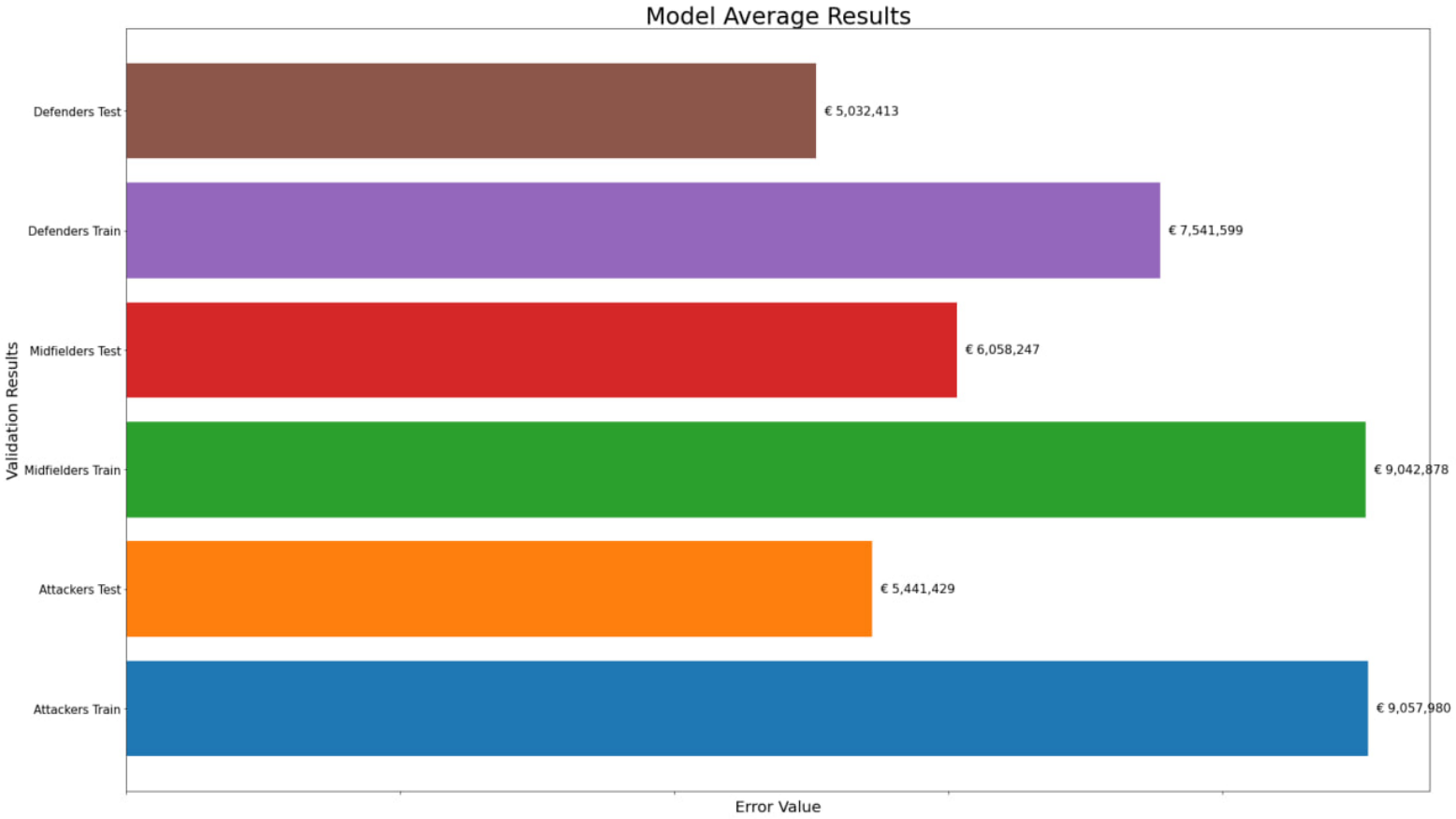


Rysunek 31: Błąd modeli wykorzystującej regresję grzbietową dla wszystkich pozycji

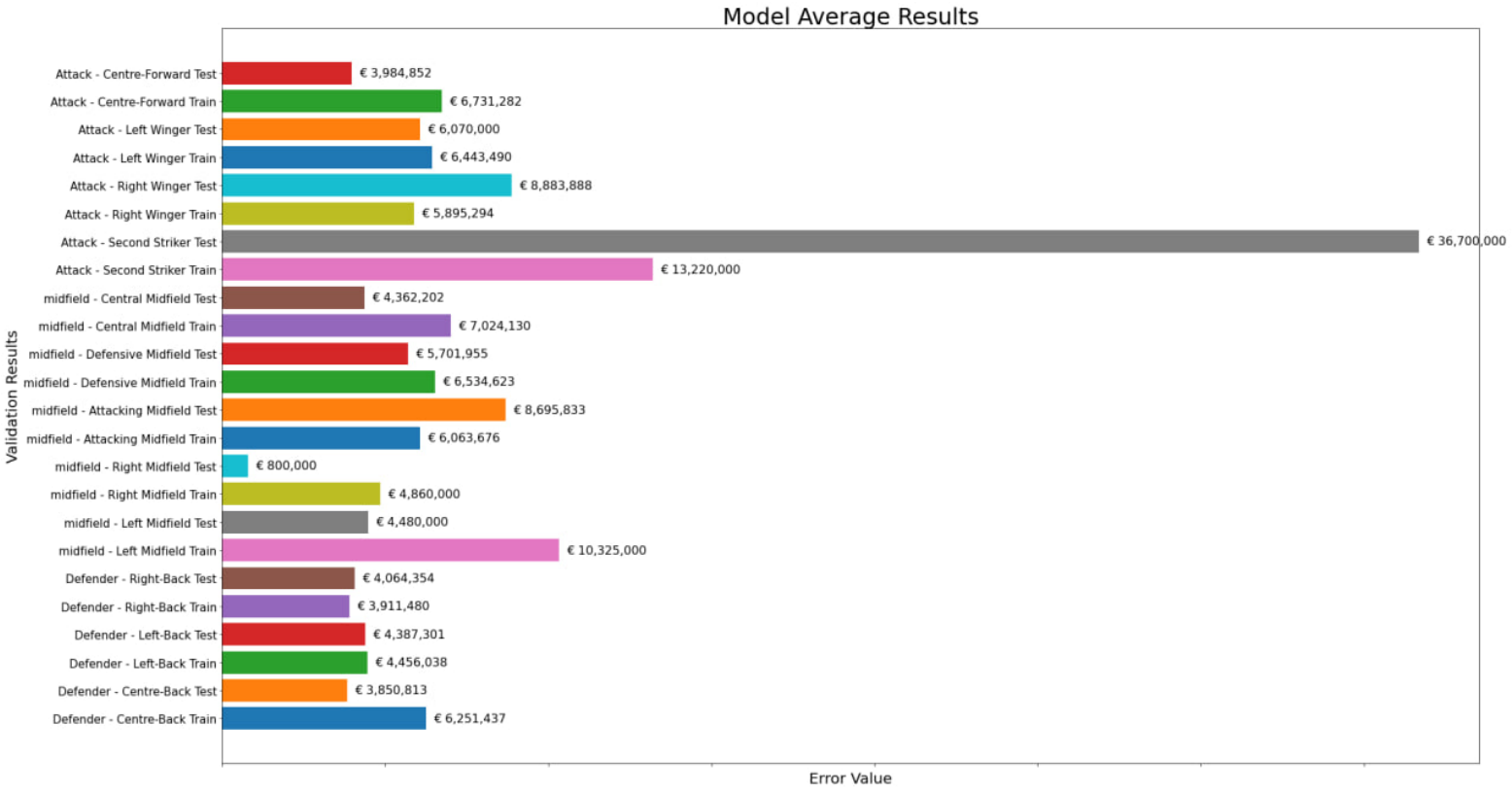


Rysunek 32: Błąd modeli wykorzystującej regresję grzbietową bez uwzględnienia pozycji

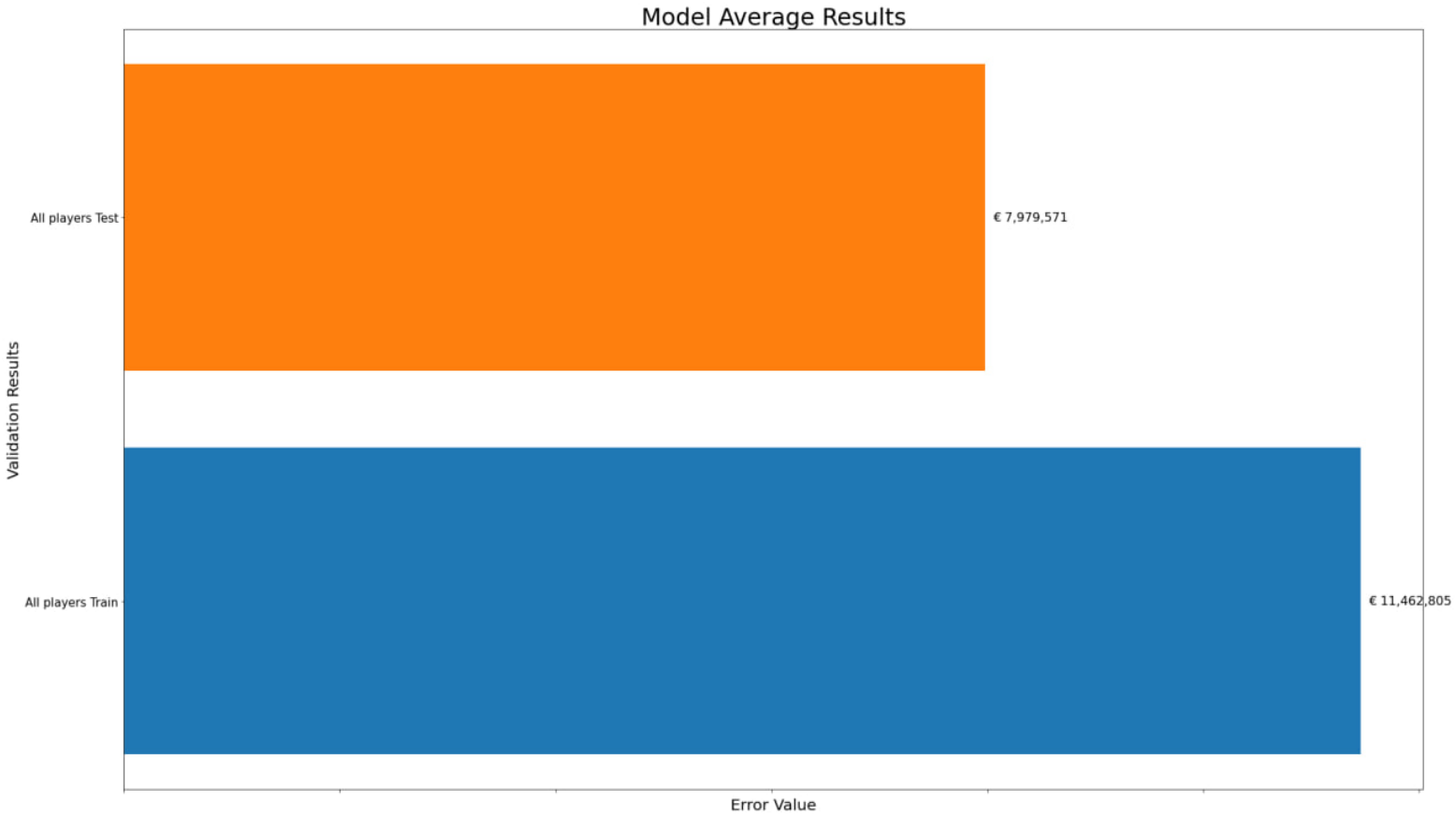
5.3.4 Rezultaty AdaBoost



Rysunek 33: Błąd modeli wykorzystującej AdaBoost dla podstawowych pozycji

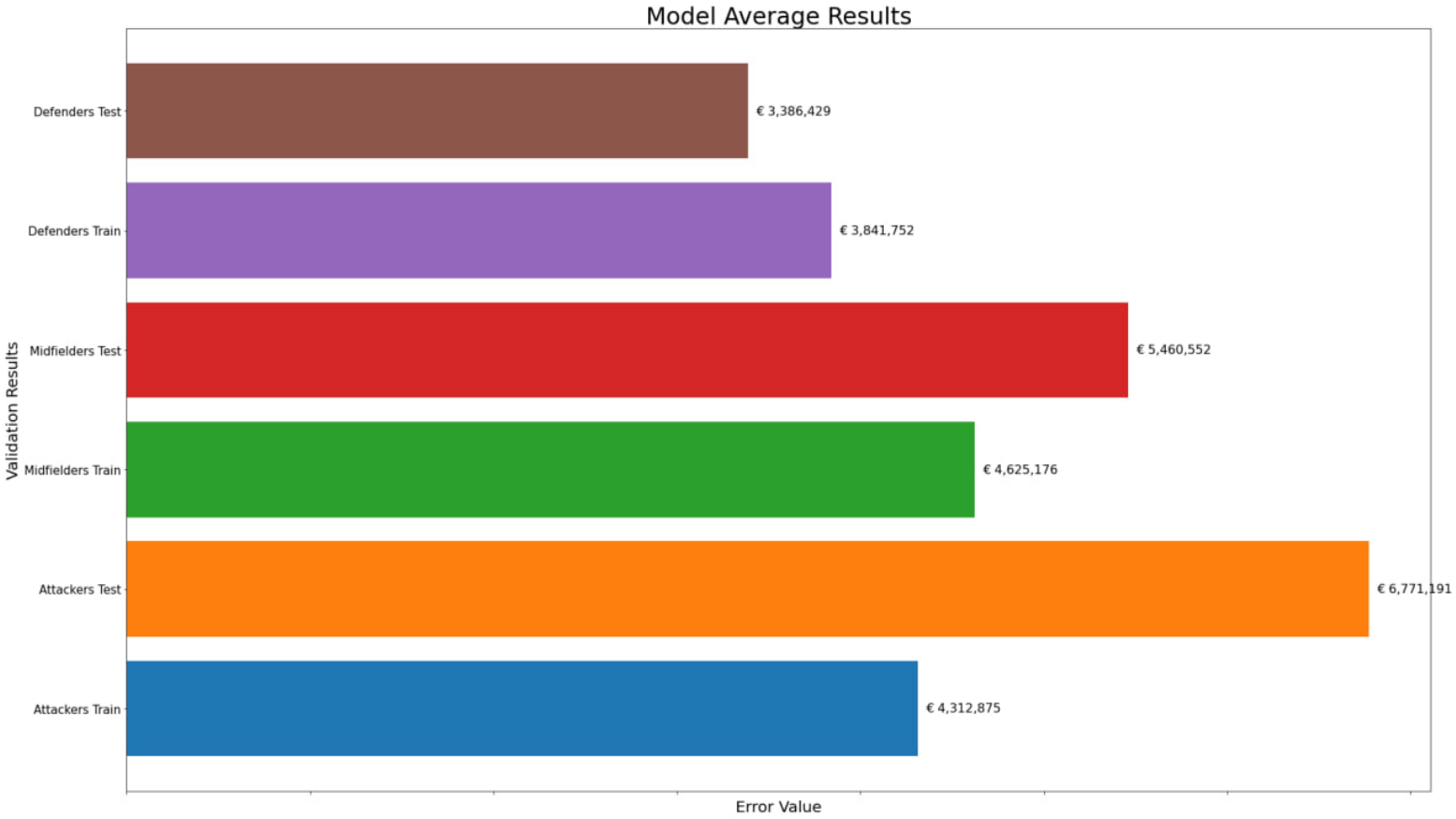


Rysunek 34: Błąd modeli wykorzystującej AdaBoost dla wszystkich pozycji

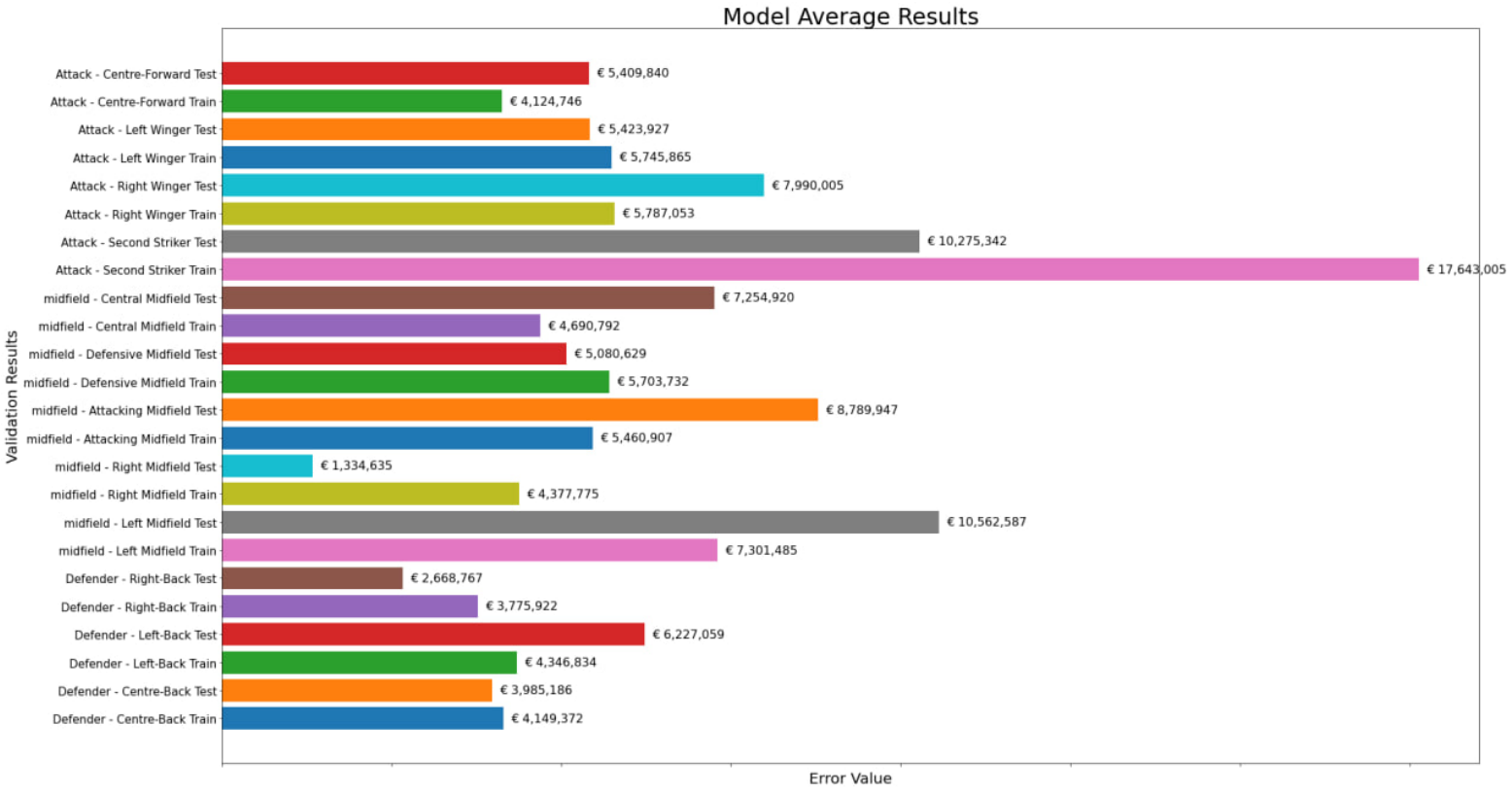


Rysunek 35: Błąd modeli wykorzystującej AdaBoost bez uwzględnienia pozycji

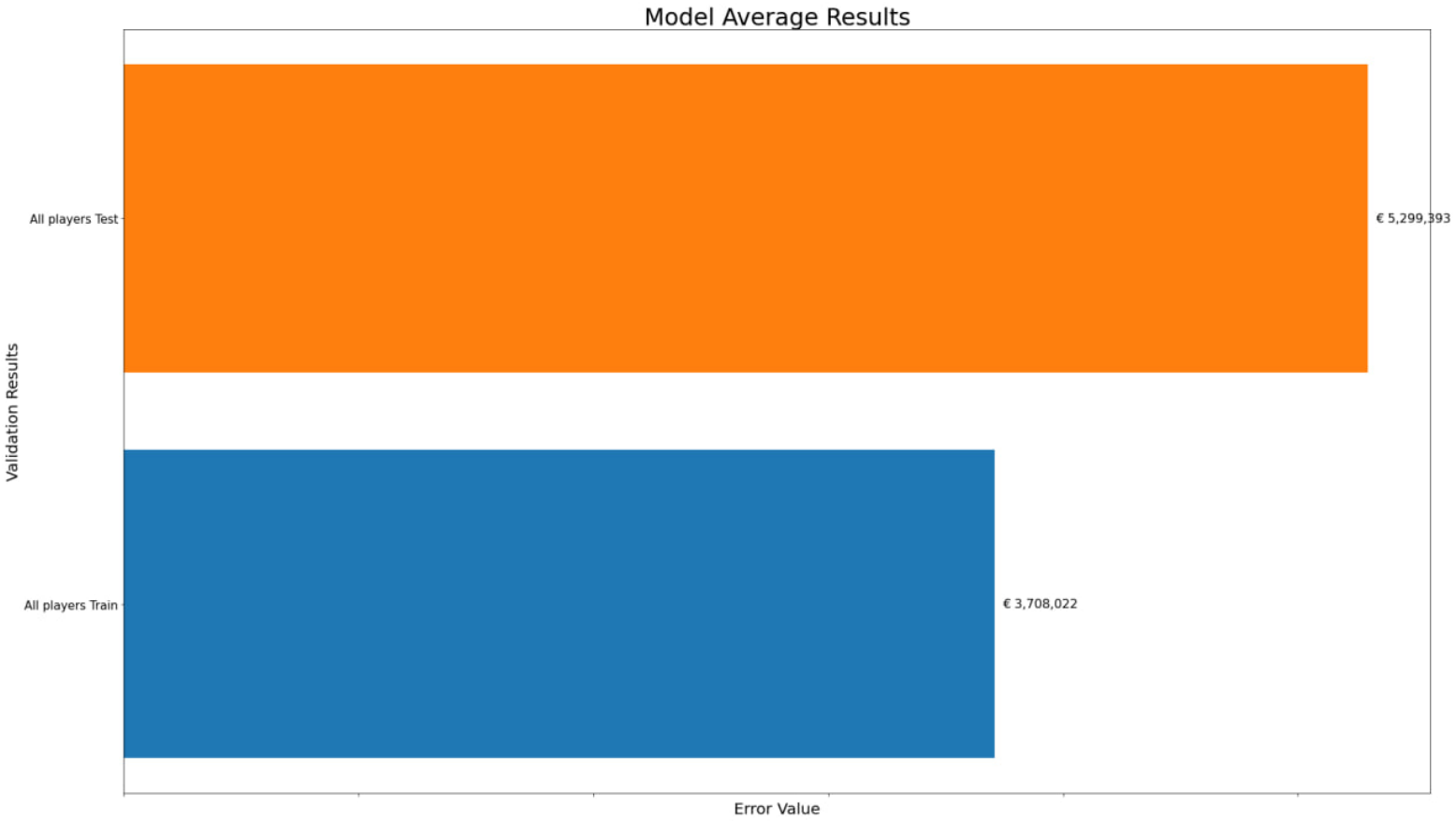
5.3.5 Rezultaty GradientBoost



Rysunek 36: Błąd modeli wykorzystującej GradientBoost dla podstawowych pozycji

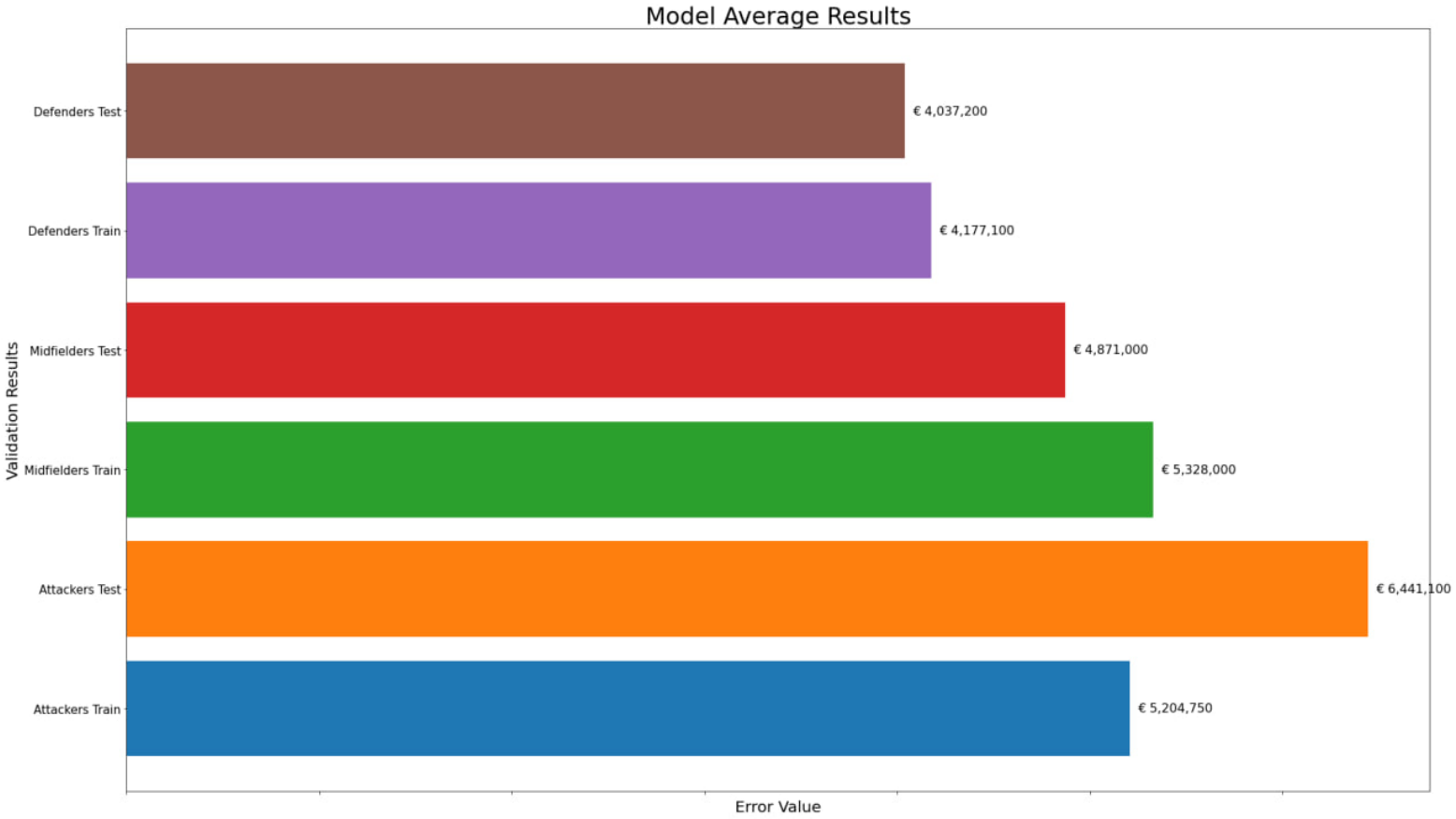


Rysunek 37: Błąd modeli wykorzystującej GradientBoost dla wszystkich pozycji

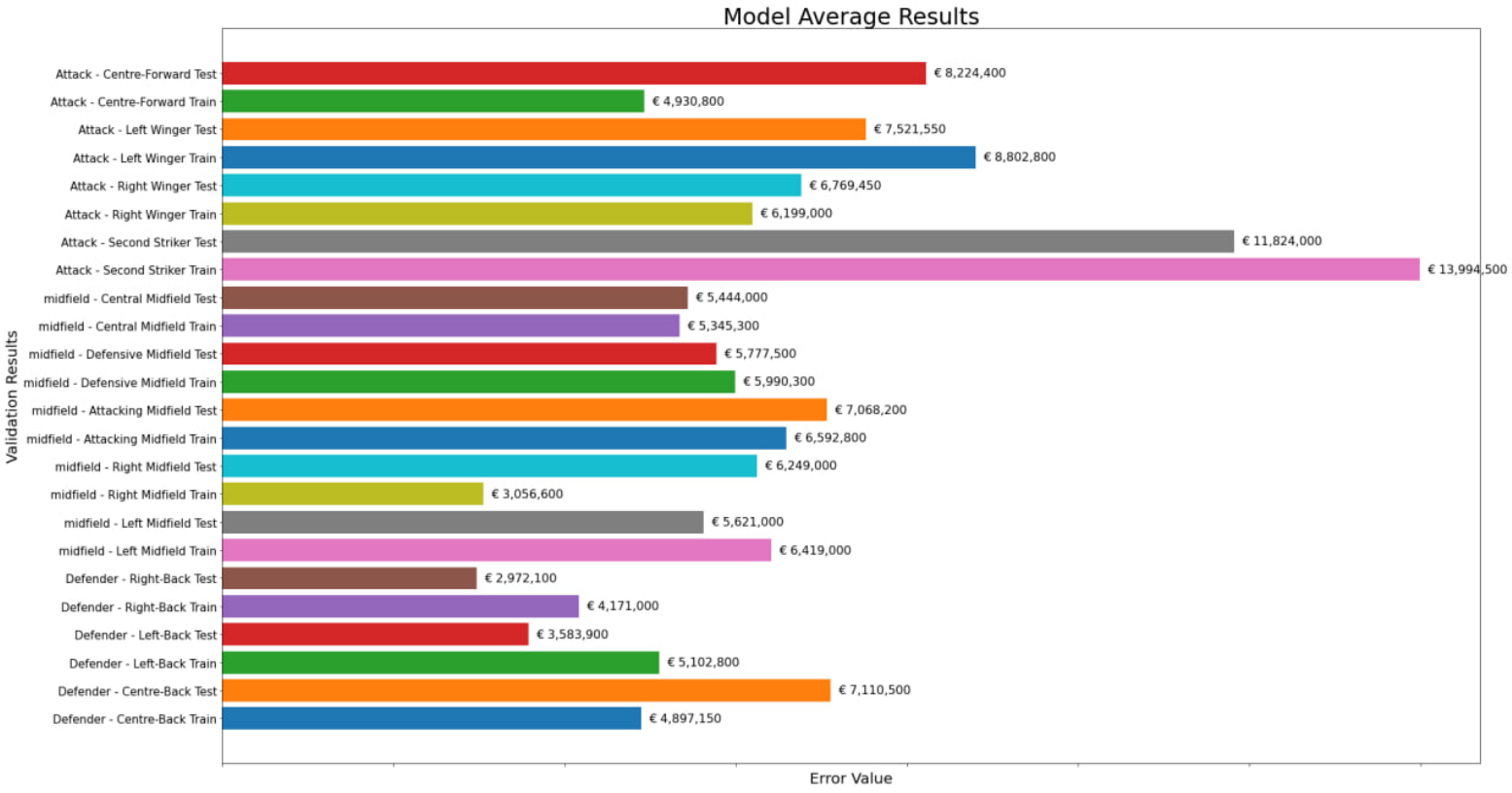


Rysunek 38: Błąd modeli wykorzystującej GradientBoost bez uwzględnienia pozycji

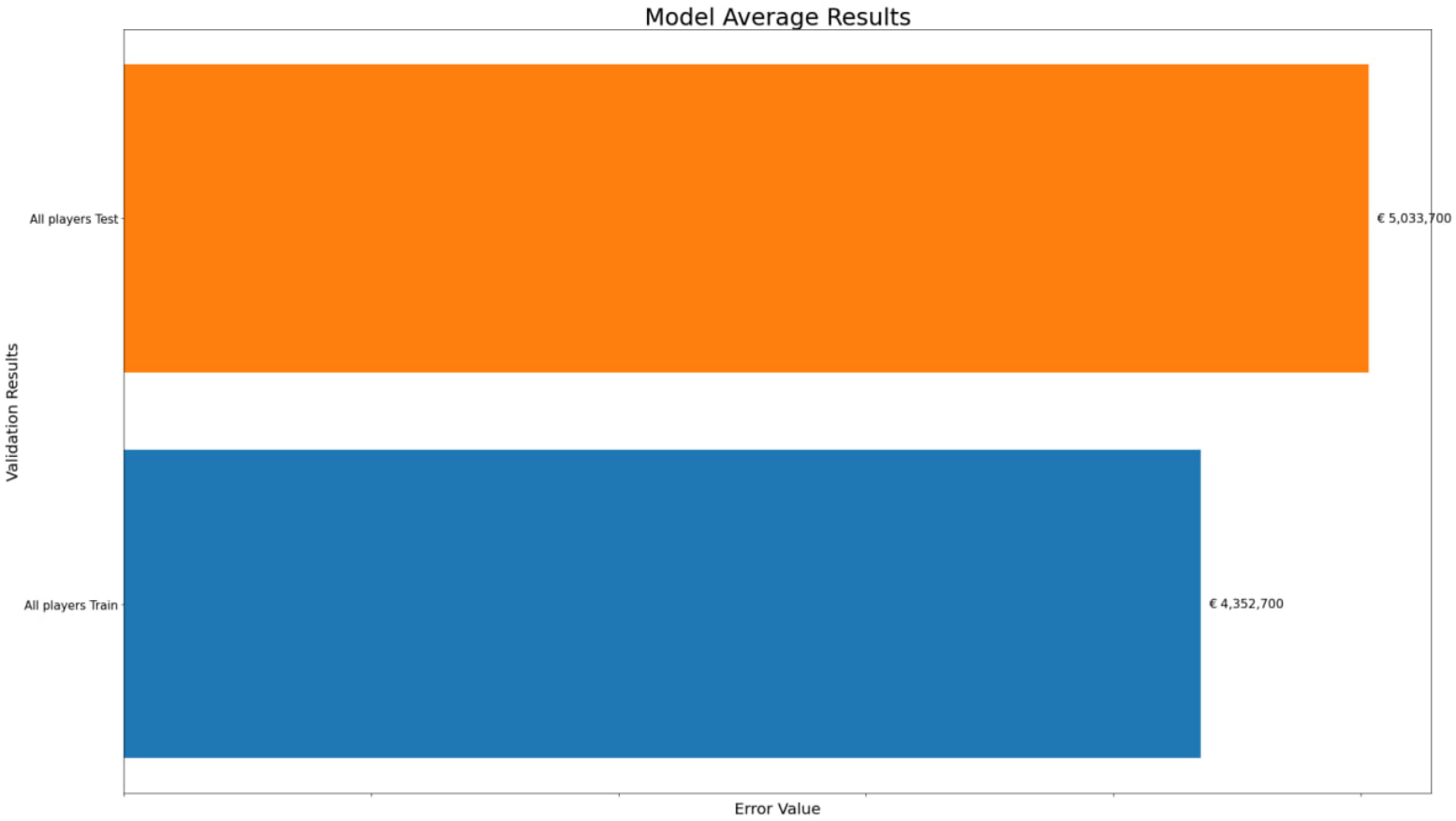
5.3.6 Rezultaty RandomForest



Rysunek 39: Błąd modeli wykorzystującej RandomForest dla podstawowych pozycji

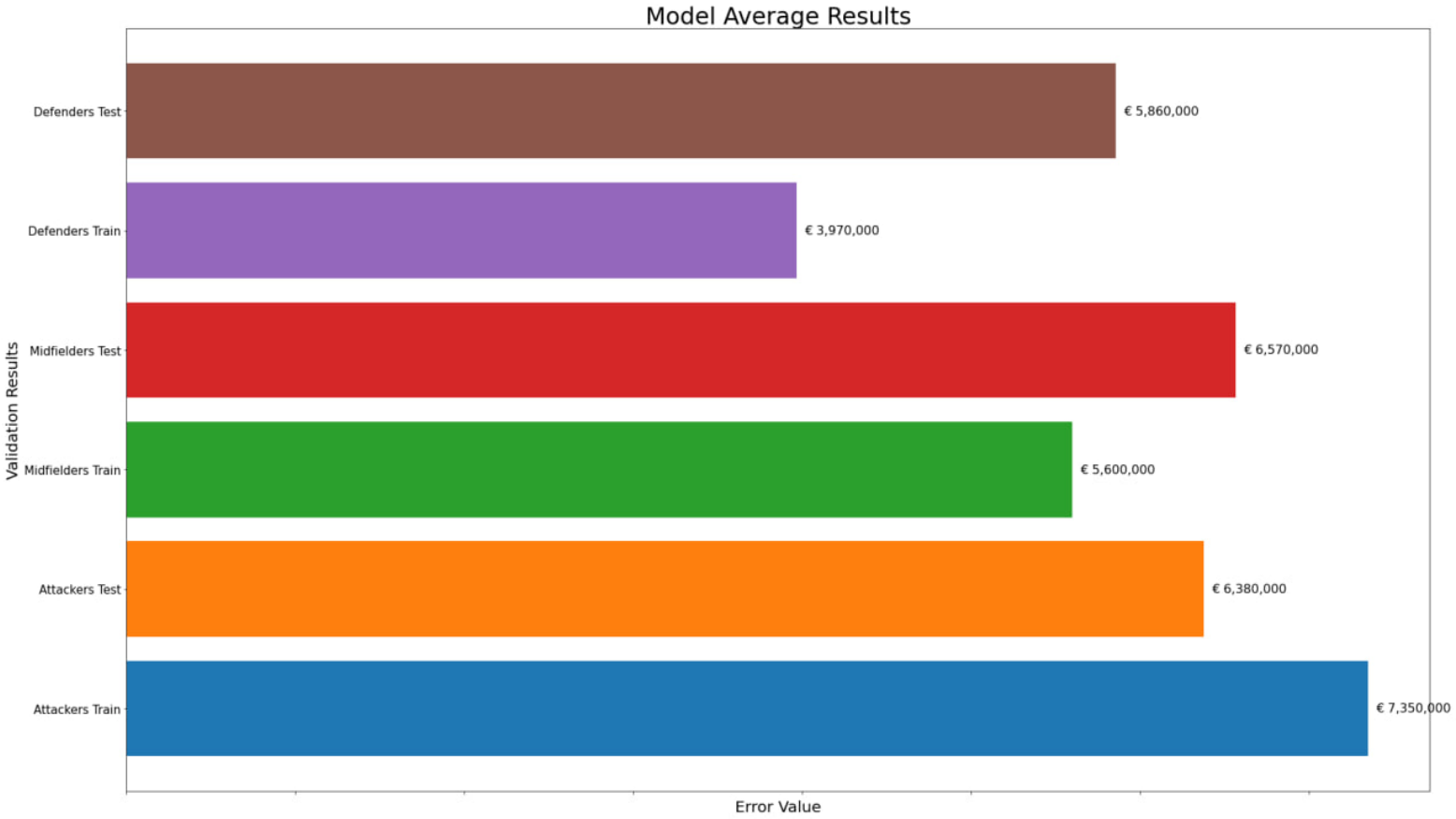


Rysunek 40: Błąd modeli wykorzystującej RandomForest dla wszystkich pozycji

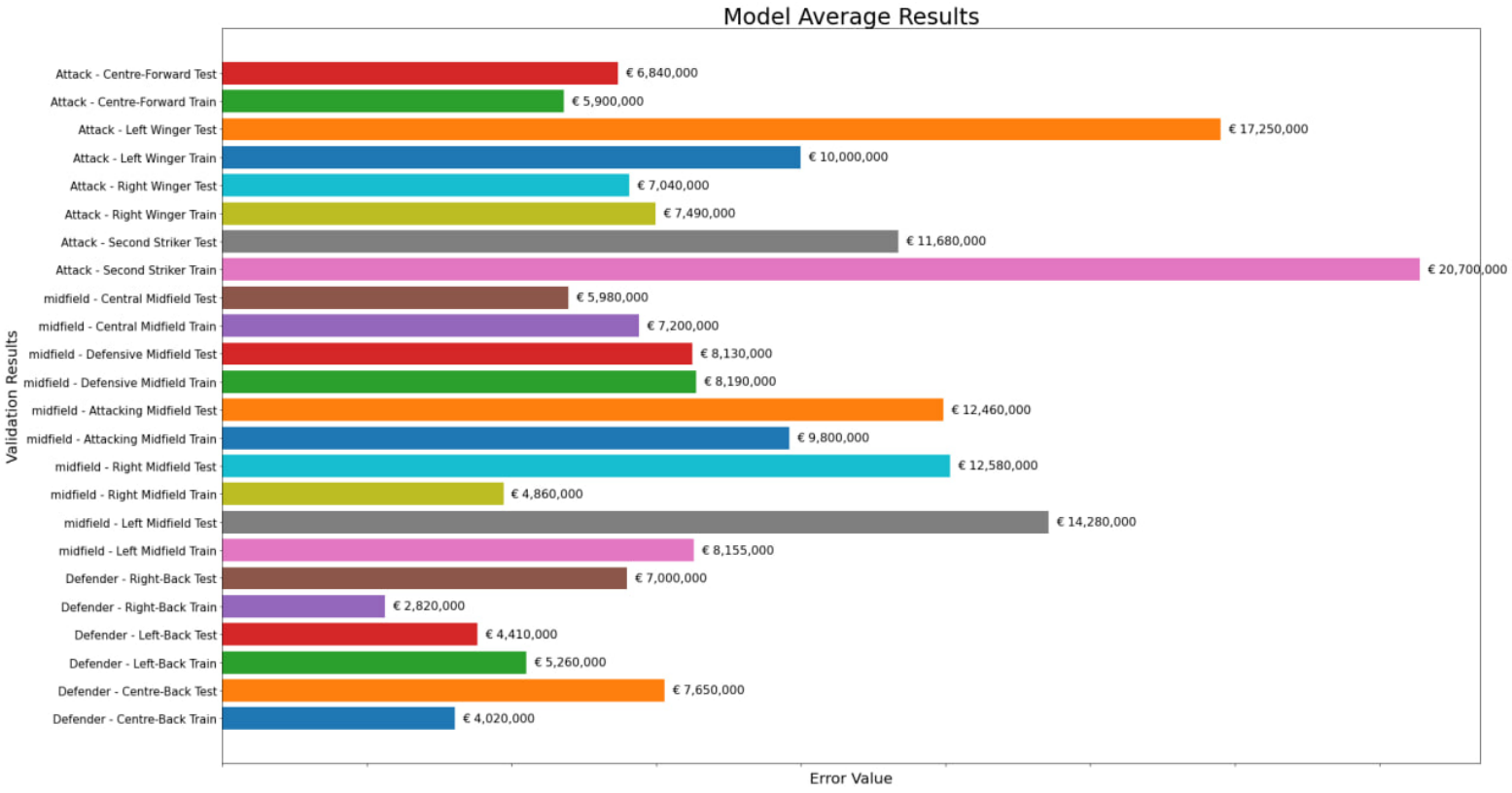


Rysunek 41: Błąd modeli wykorzystującej RandomForest bez uwzględnienia pozycji

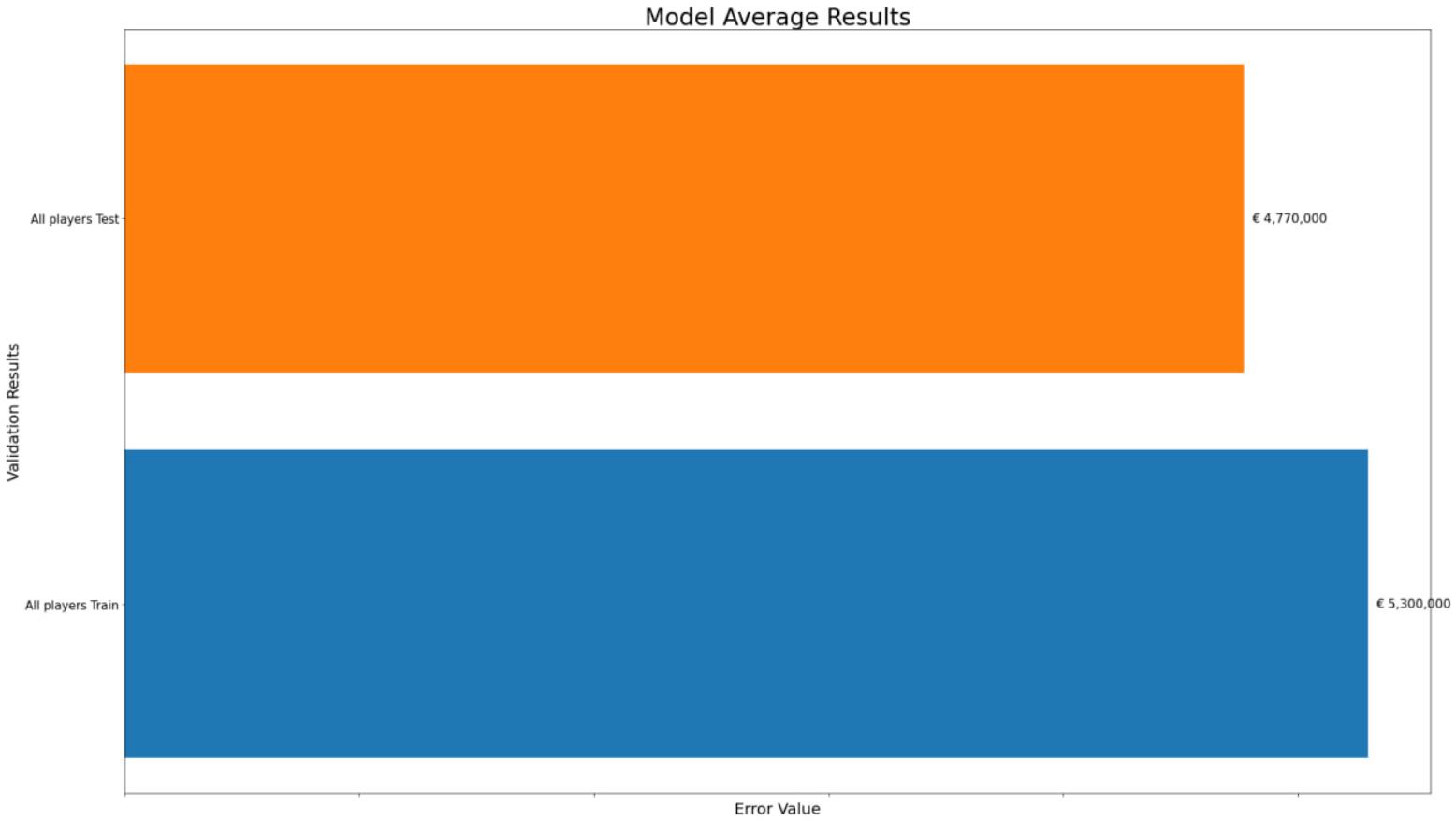
5.3.7 Rezultaty DecisionTree



Rysunek 42: Błąd modeli wykorzystującej DecisionTree dla podstawowych pozycji



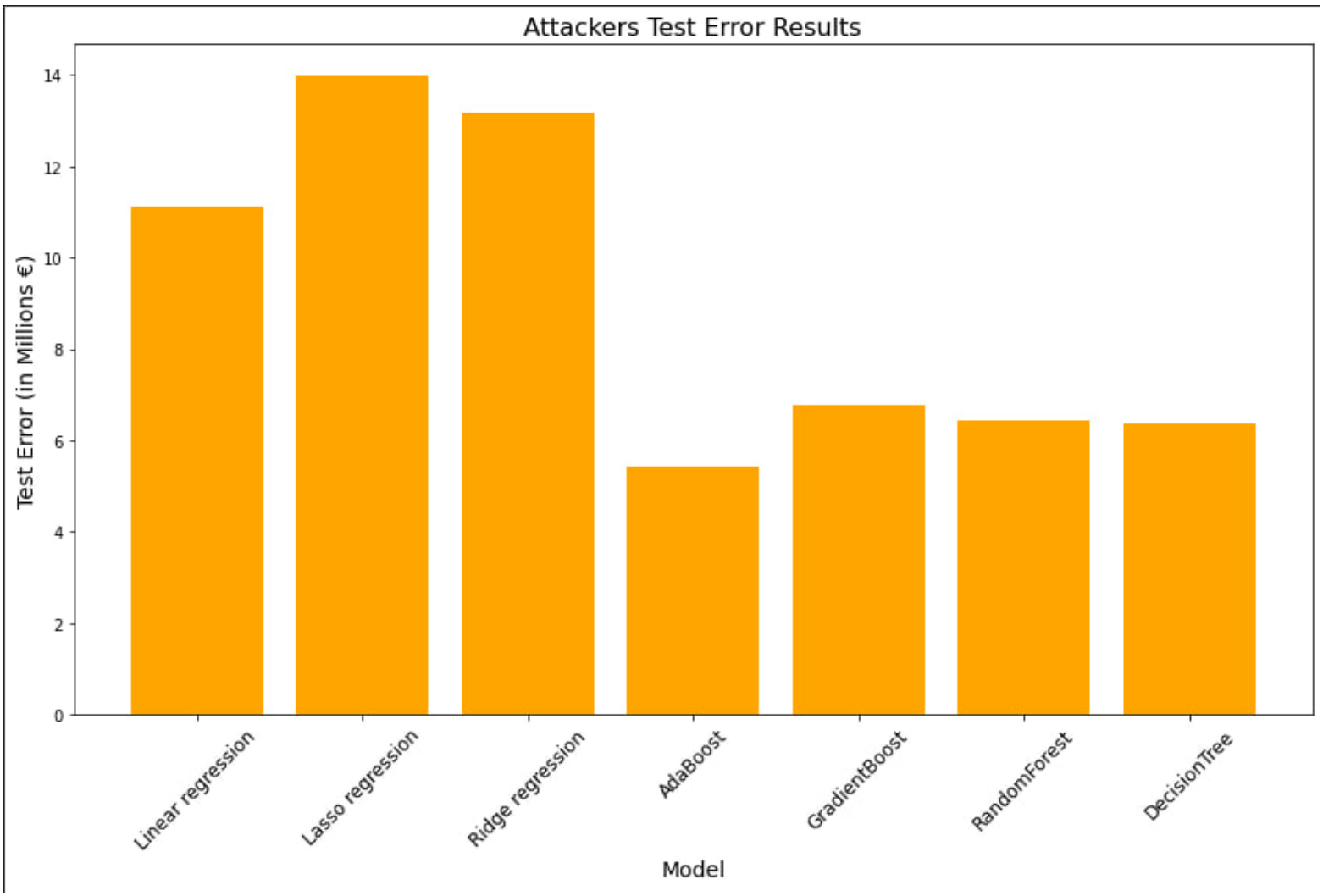
Rysunek 43: Błąd modeli wykorzystującej DecisionTree dla wszystkich pozycji



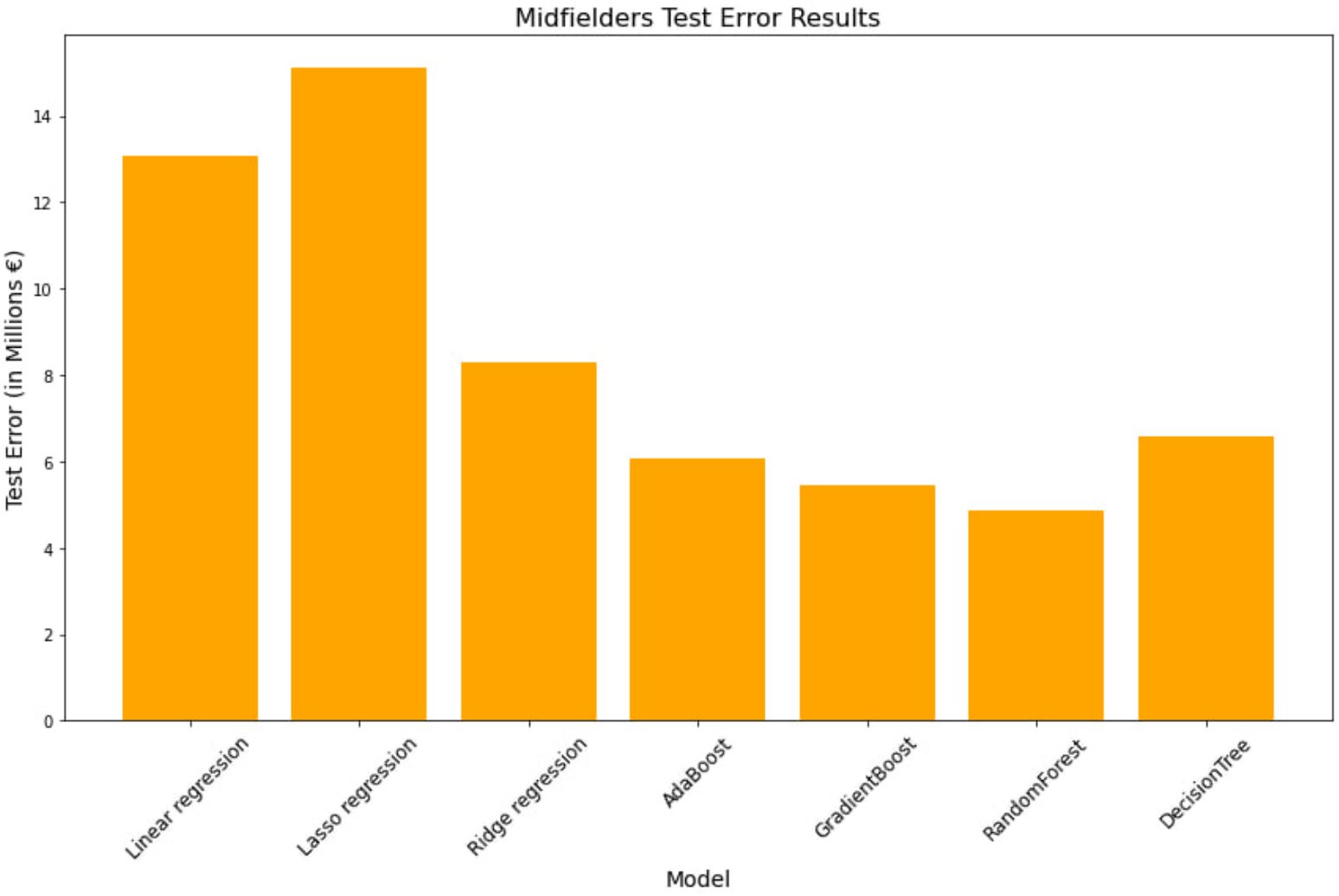
Rysunek 44: Błąd modeli wykorzystującej DecisionTree bez uwzględnienia pozycji

5.4 Porównanie modeli

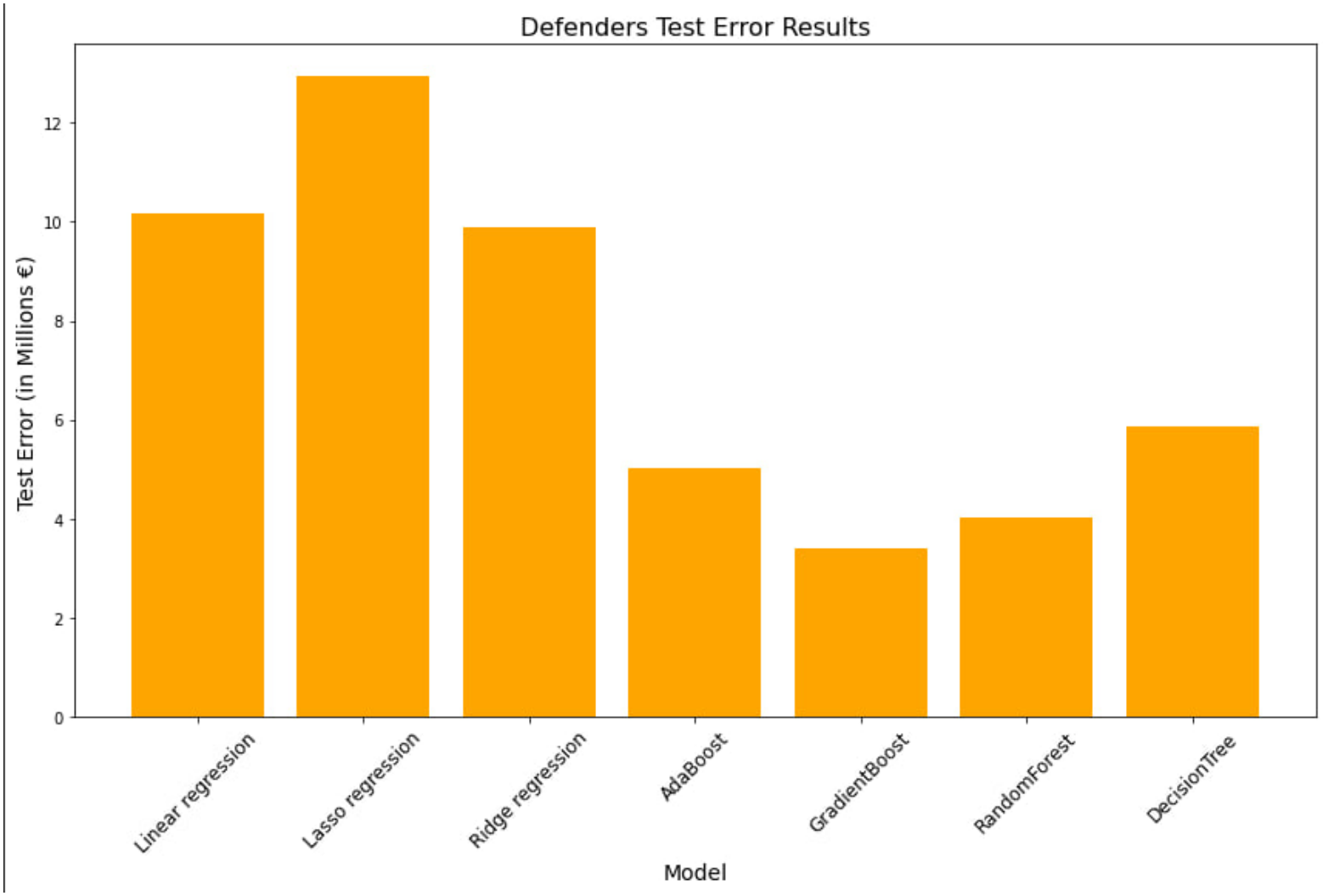
5.4.1 Porównanie modeli dla podstawowych pozycji



Rysunek 45: Porównanie modeli dla napastników

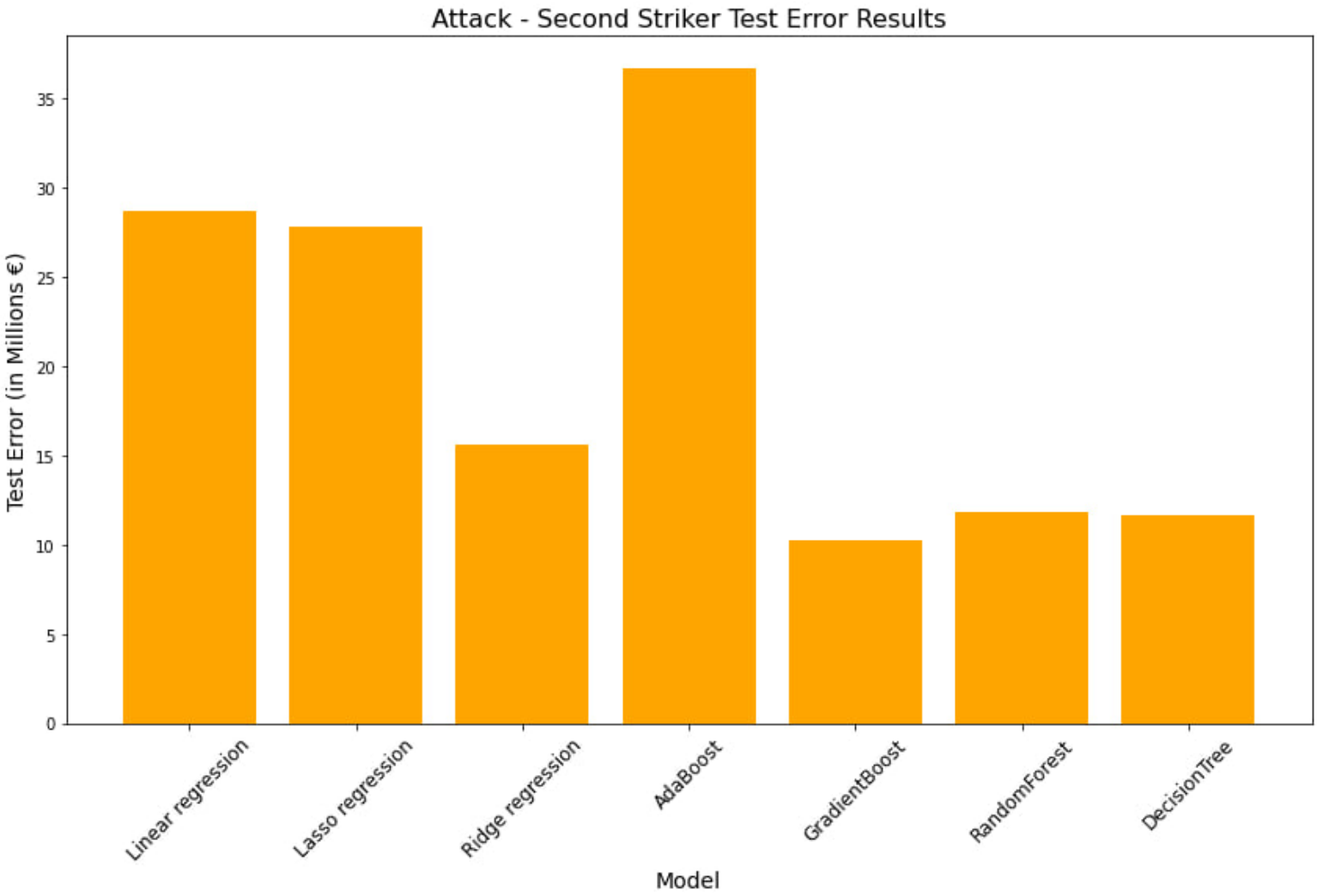


Rysunek 46: Porównanie modeli dla pomocników

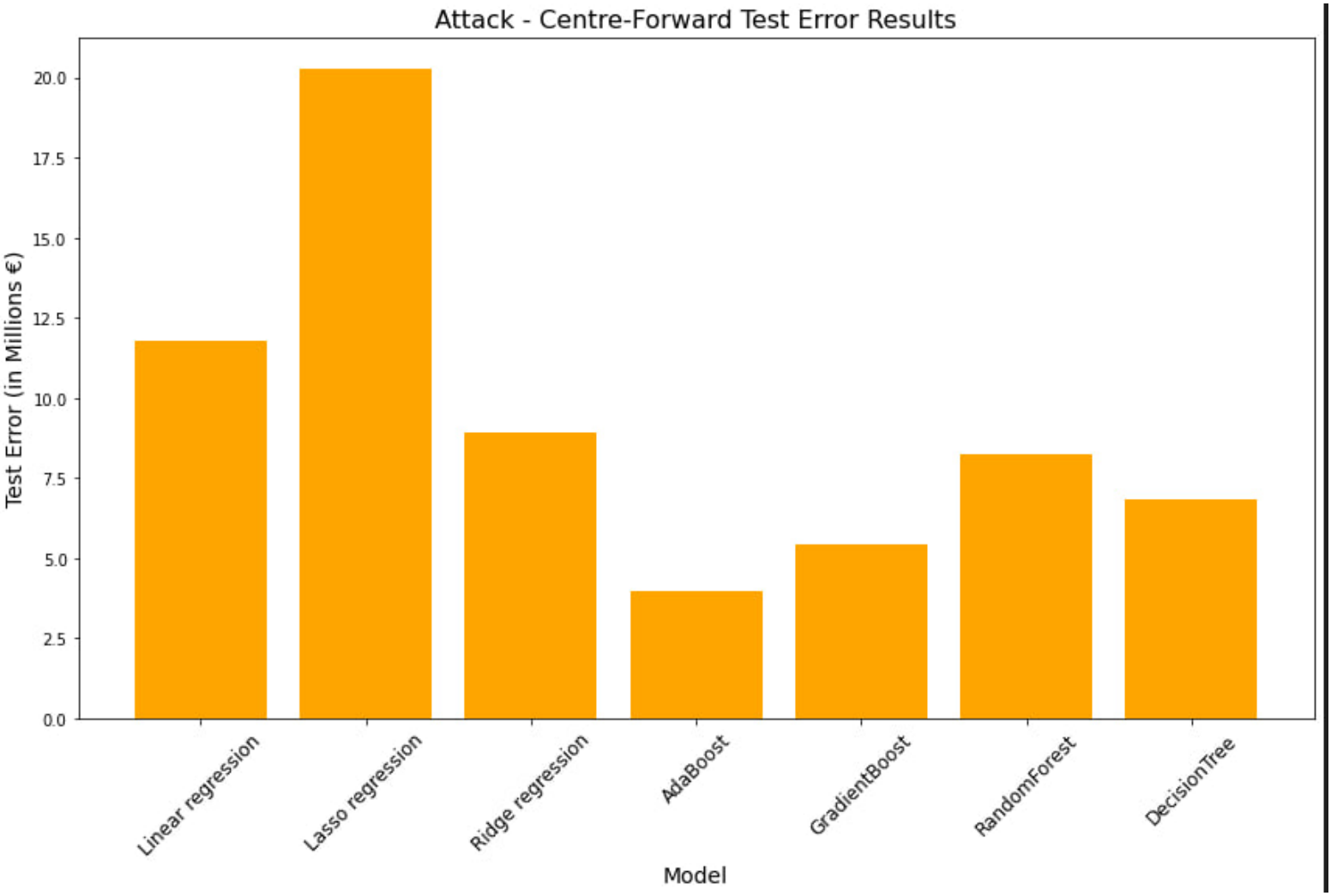


Rysunek 47: Porównanie modeli dla obrońców

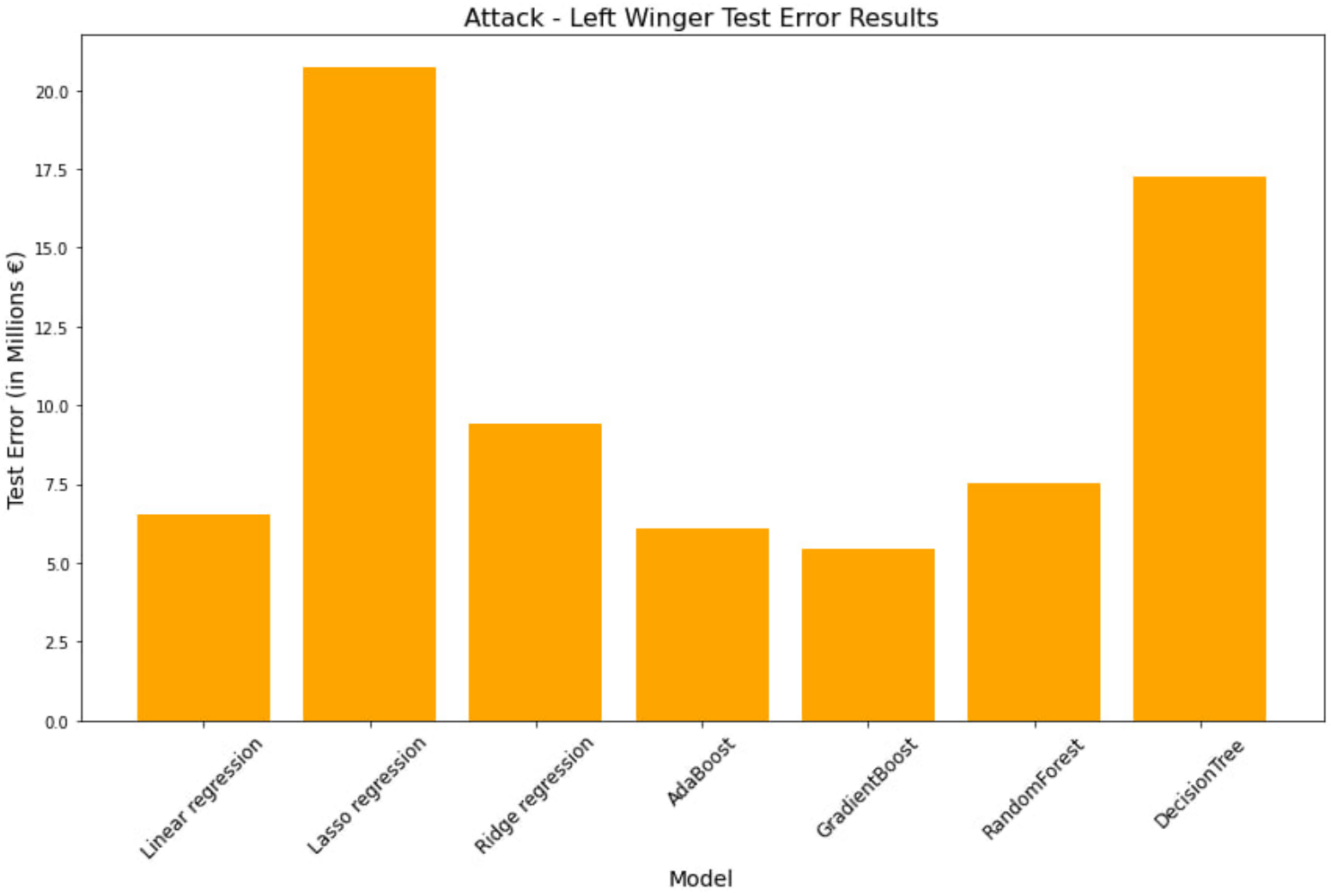
5.4.2 Porównanie modeli dla wszystkich pozycji



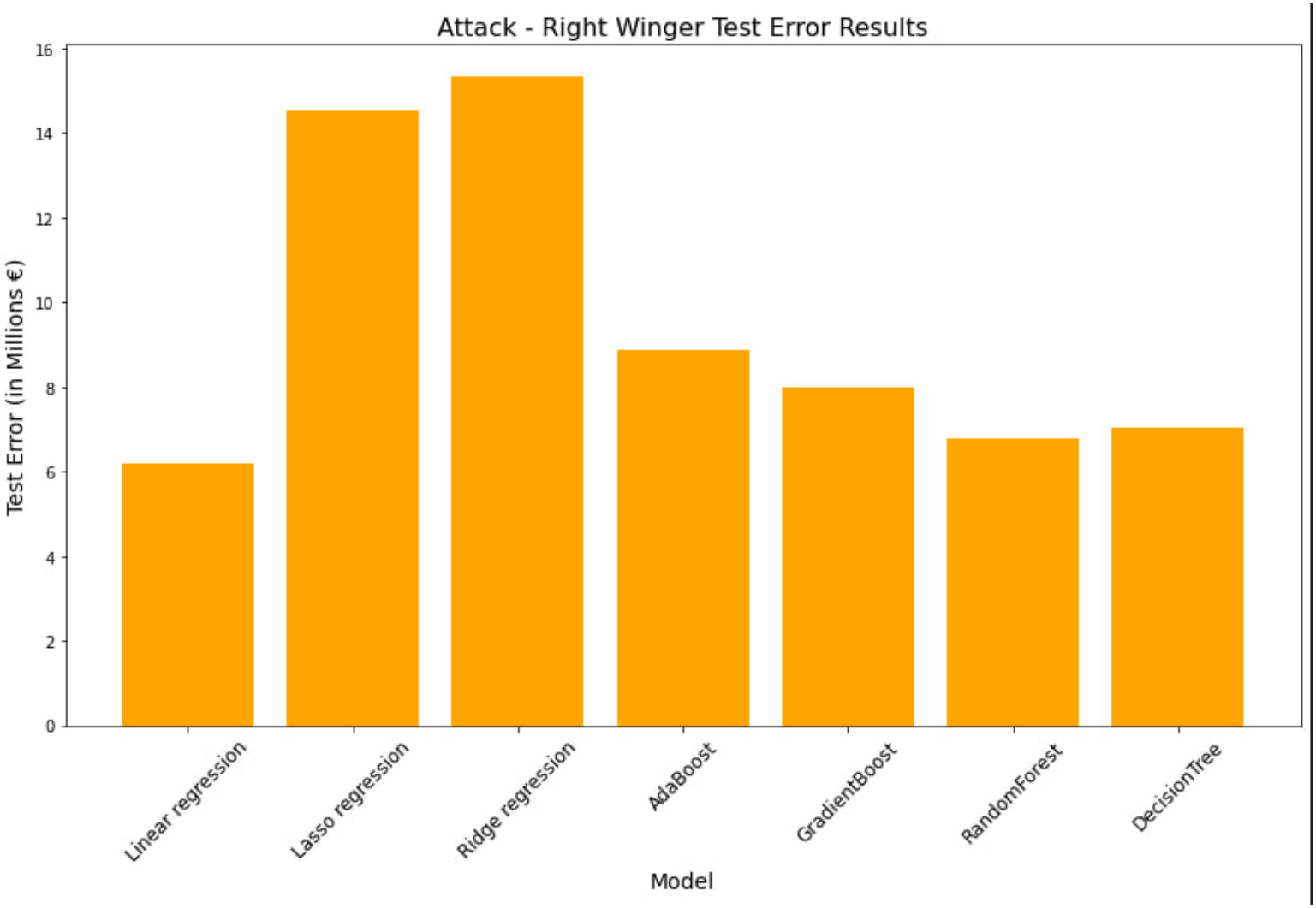
Rysunek 48: Porównanie modeli dla drugich napastników



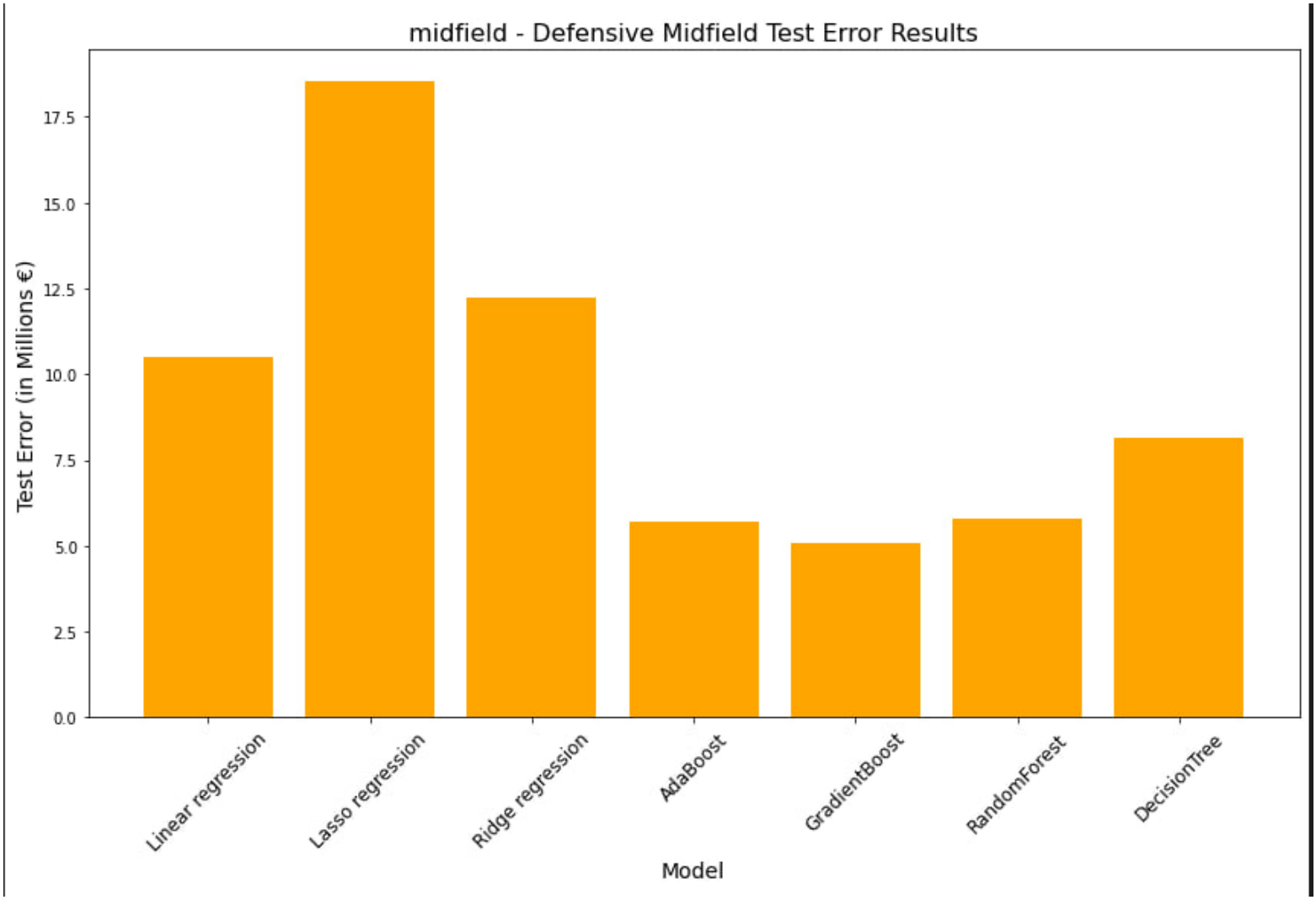
Rysunek 49: Porównanie modeli dla centr-forwardów



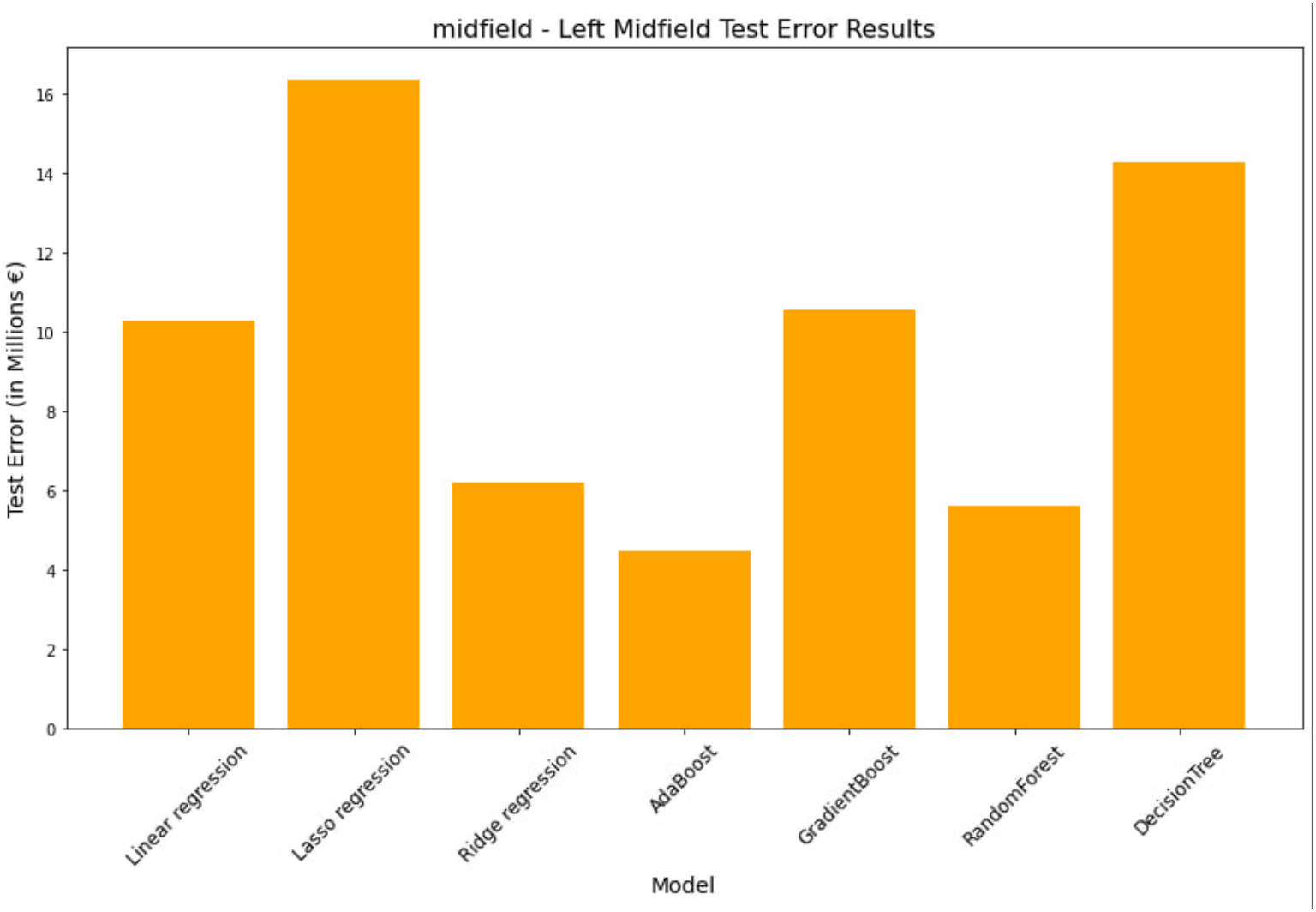
Rysunek 50: Porównanie modeli dla lewych wingerów



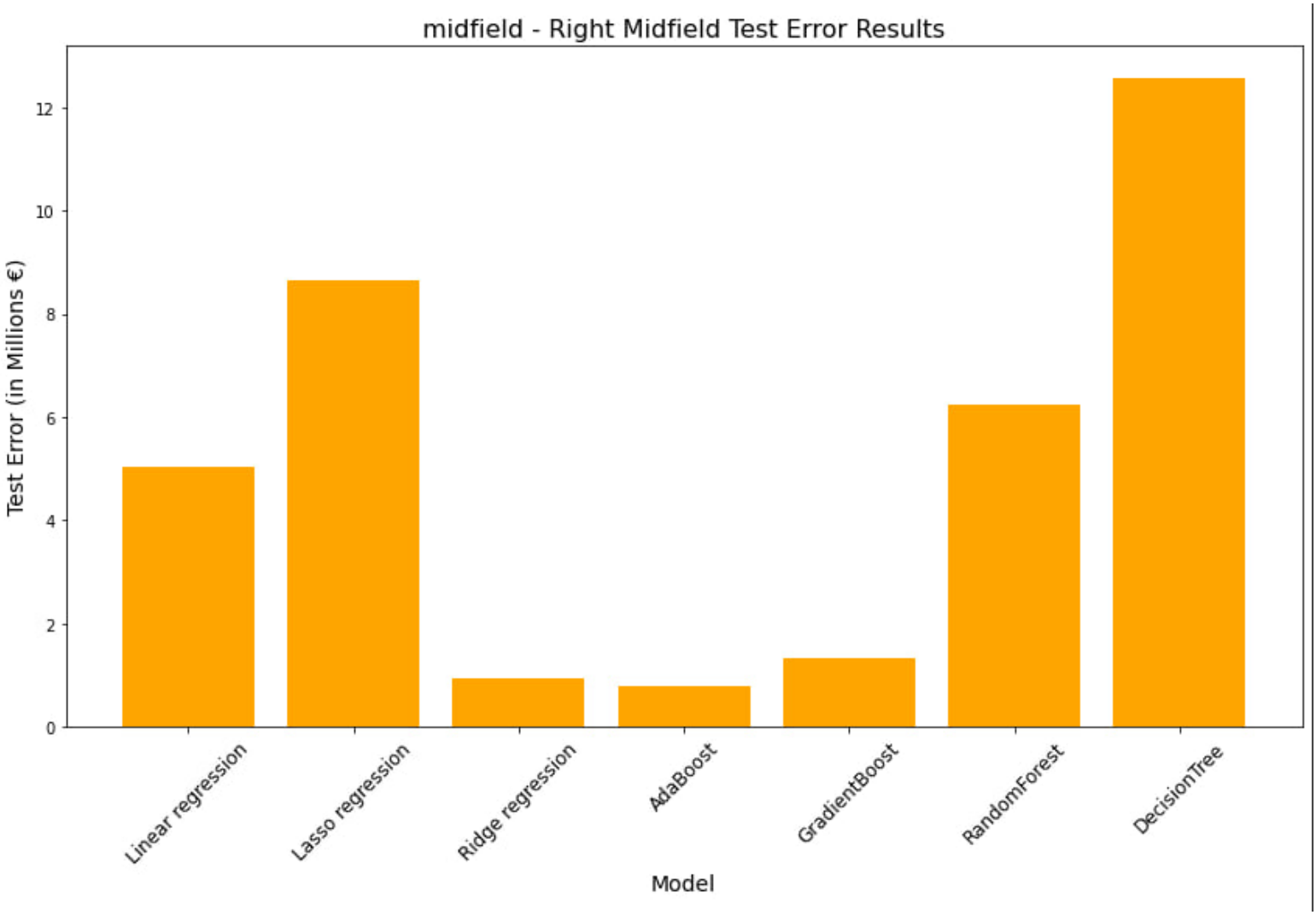
Rysunek 51: Porównanie modeli dla prawych wingerów



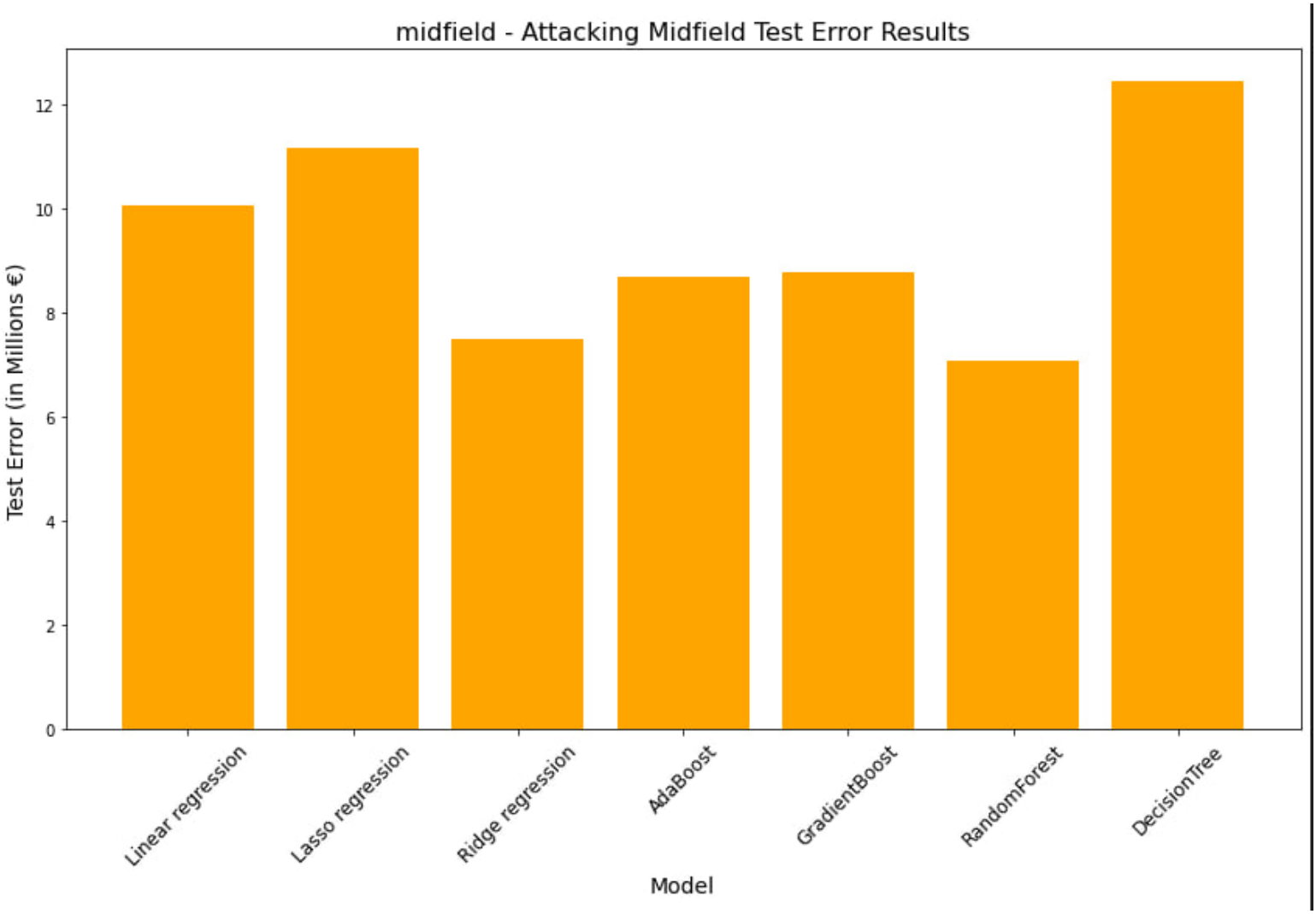
Rysunek 52: Porównanie modeli dla defensywnych pomocników



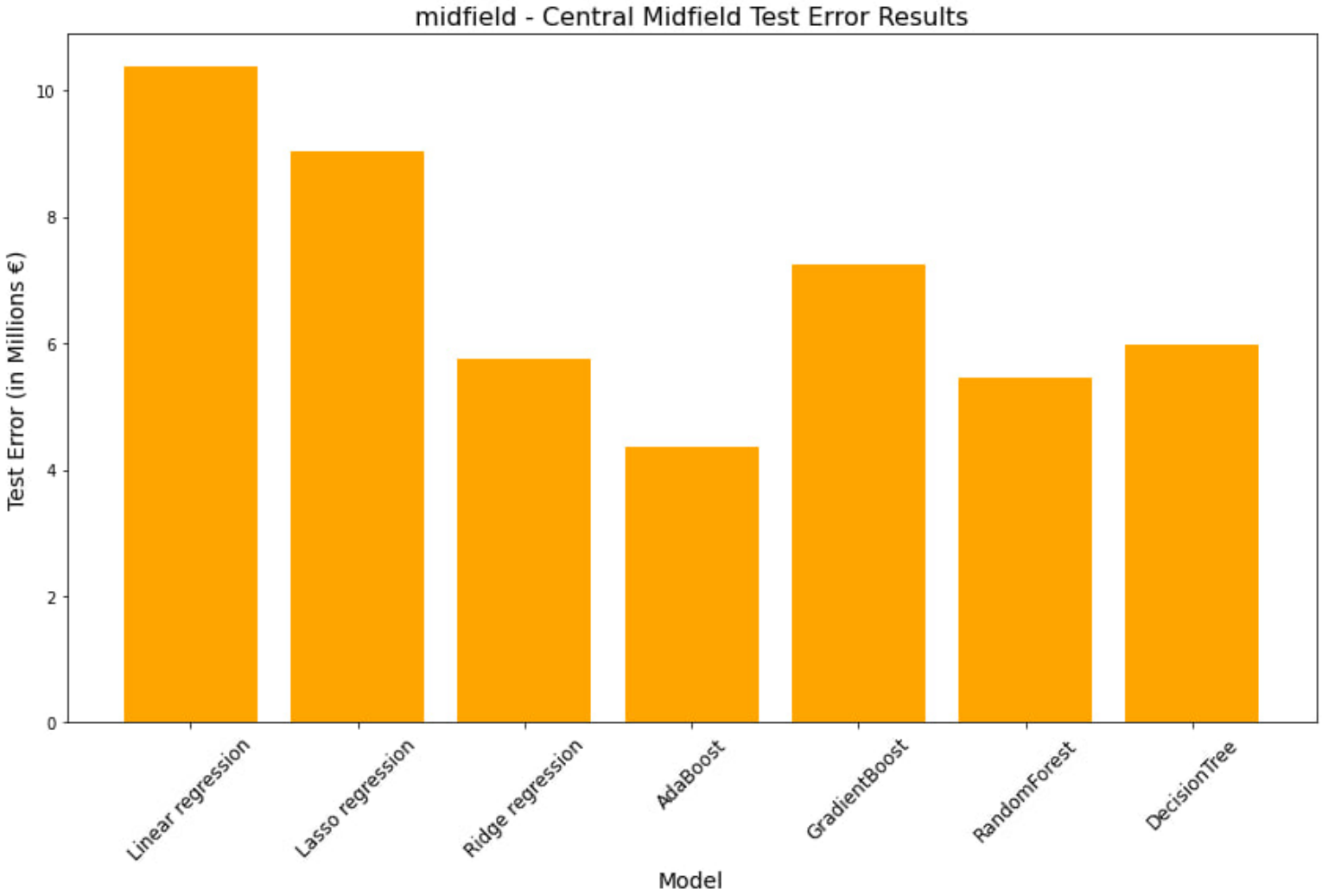
Rysunek 53: Porównanie modeli dla lewych pomocników



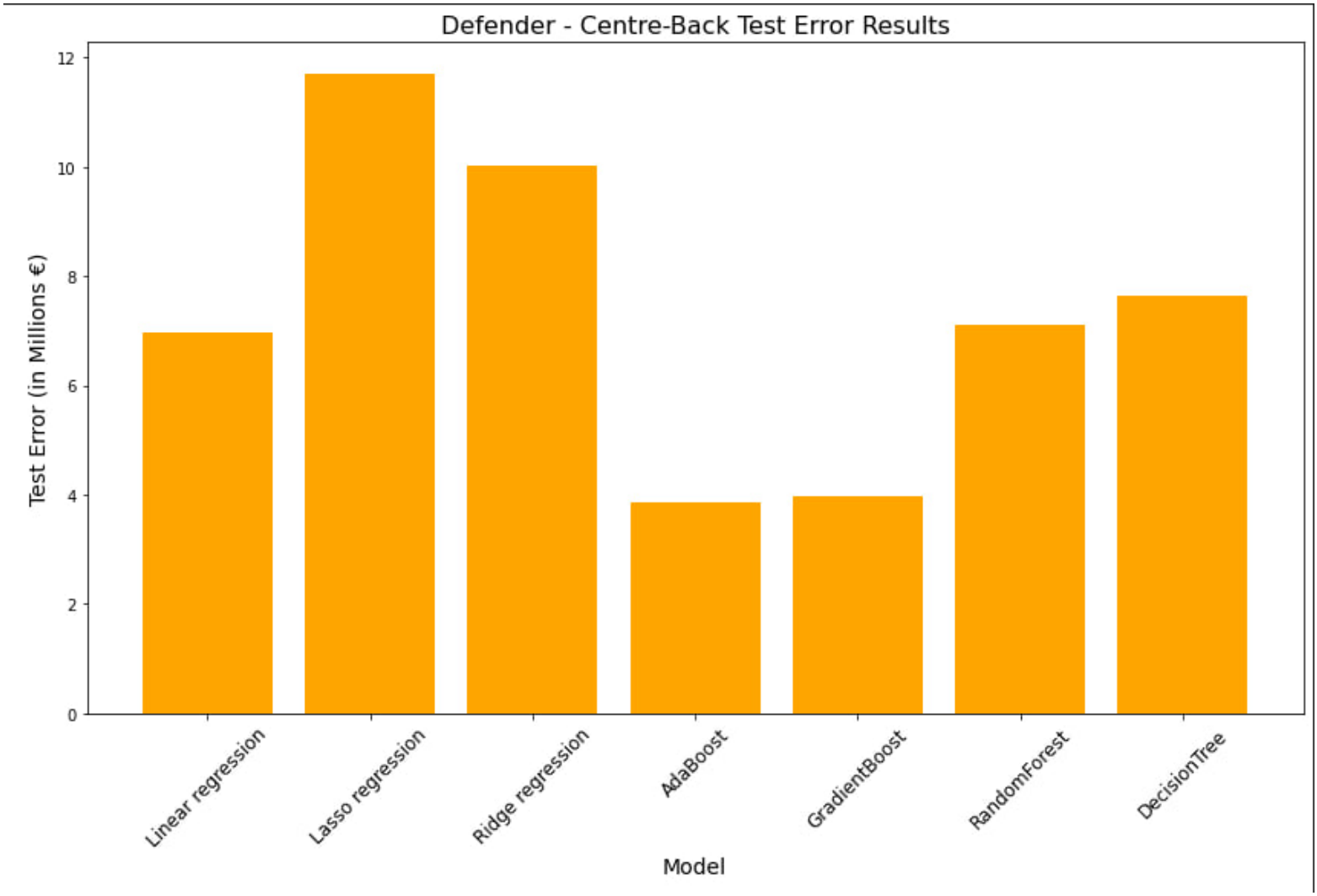
Rysunek 54: Porównanie modeli dla prawych pomocników



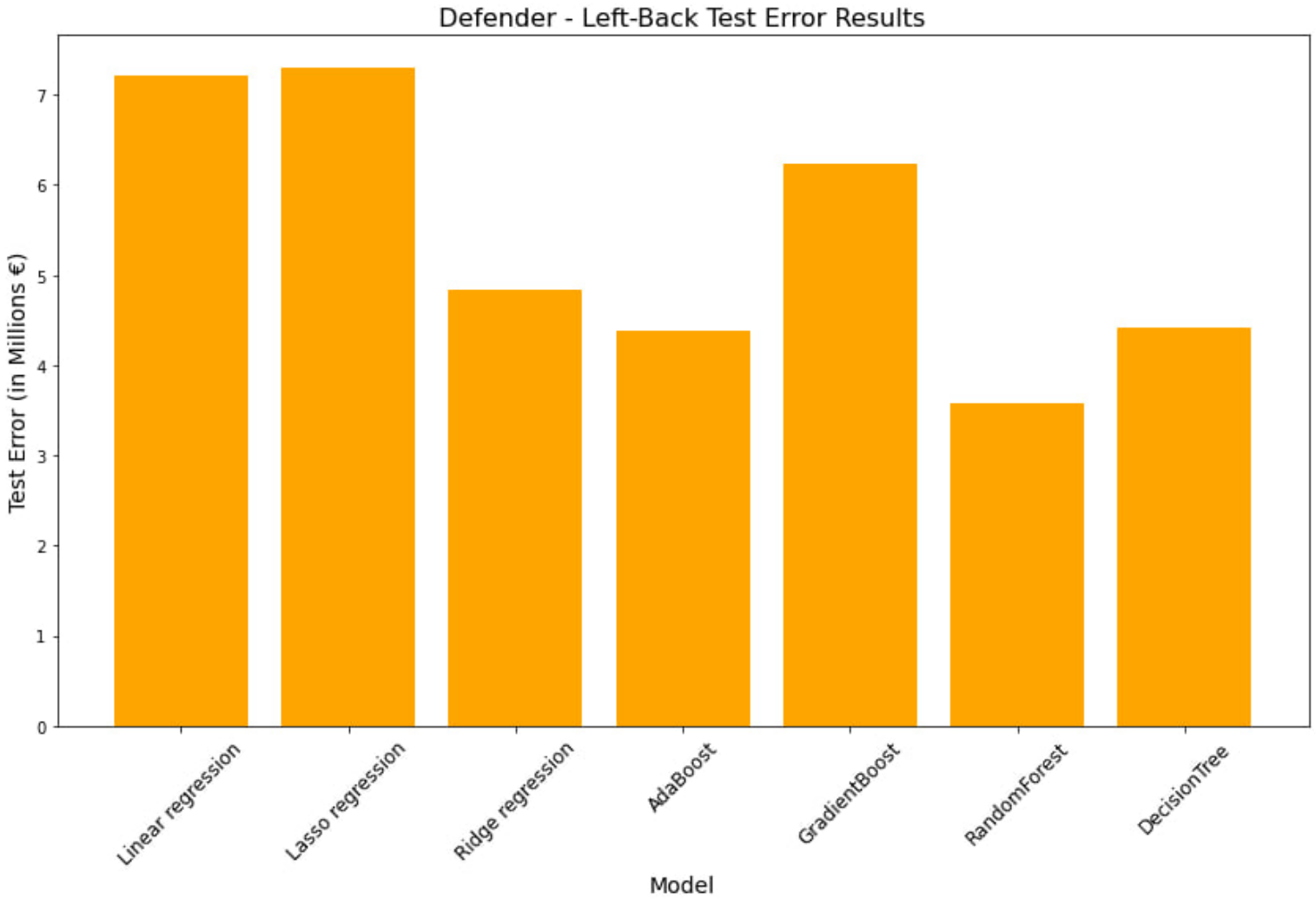
Rysunek 55: Porównanie modeli dla atakujących pomocników



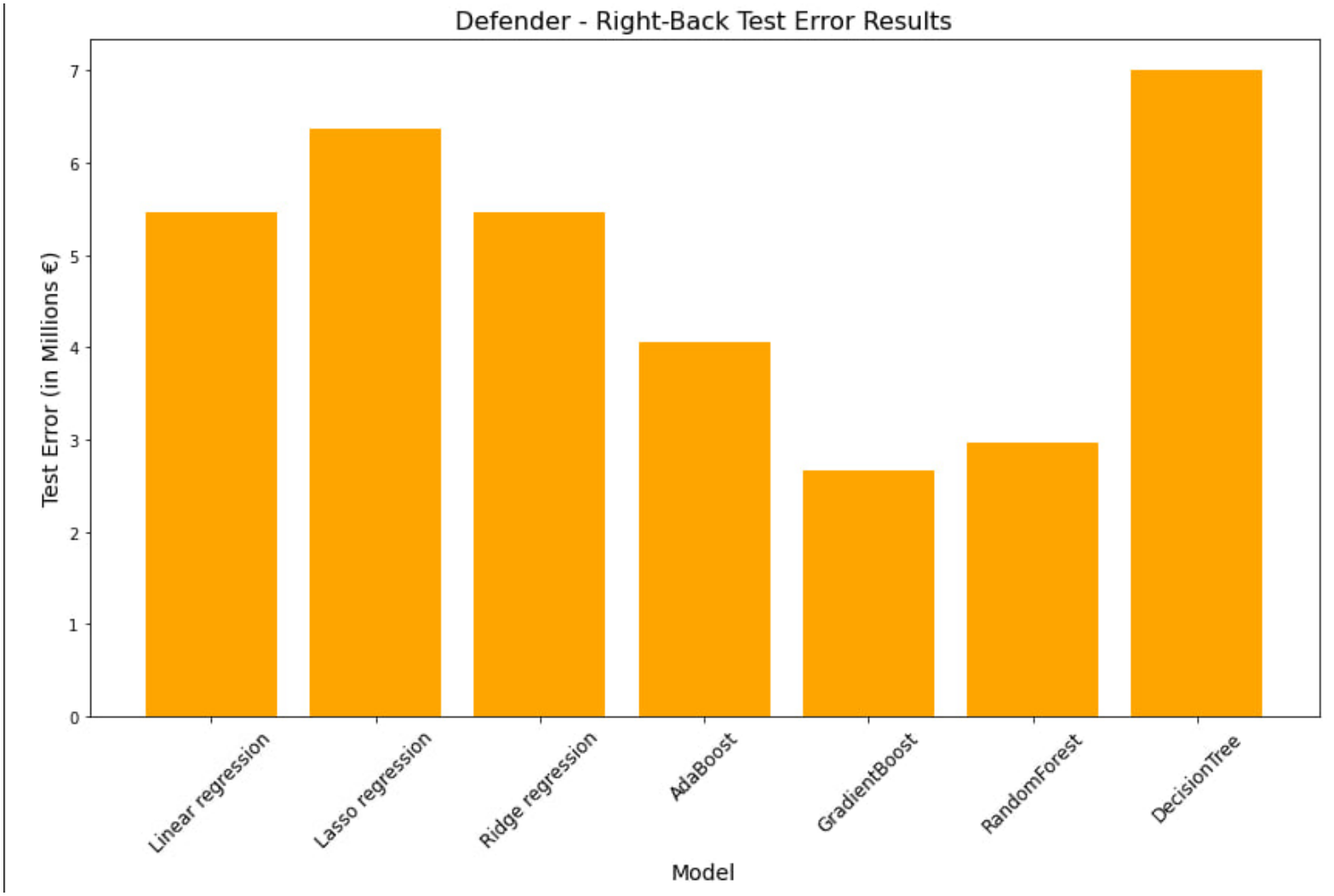
Rysunek 56: Porównanie modeli dla centralnych pomocników



Rysunek 57: Porównanie modeli dla centralnych obrońców

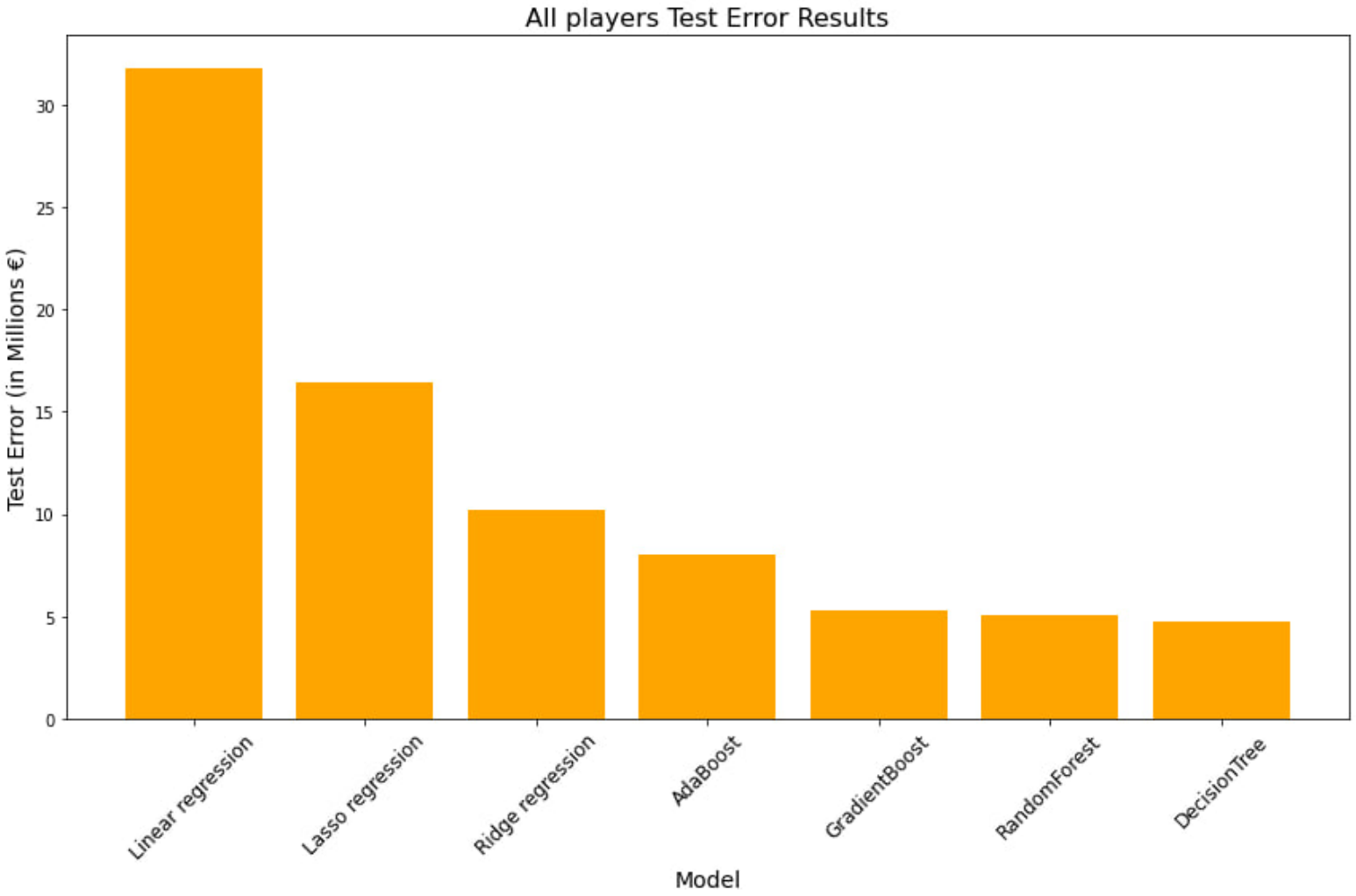


Rysunek 58: Porównanie modeli dla lewych obrońców



Rysunek 59: Porównanie modeli dla prawych obrońców

5.4.3 Porównanie modeli bez uwzględnienia pozycji



Rysunek 60: Porównanie modeli bez uwzględnienia pozycji

5.5 Wnioski związane z porównywaniem modeli

Na podstawie wykresów porównujących modeli można powiedzieć, że regresja liniowa, lasso oraz grzbietowa ogólnie mają większy błąd od bardziej zaawanoswanych modeli, takich jak AdaBoost, GradientBoost, RandomForest oraz DecisionTree (który tak naprawdę jest gorszy od 3 poprzednich) chociaż nie jest to zawsze prawda (np. błąd regresji grzbietowej dla prawych pomocników jest jednym z najlepszych, a DecisionTree - najgorszy)

Oprócz tego można też powiedzieć, że wiedza na jakiej pozycji piłkarz występuje nie bardzo wpływa na przewidywanie wartości transferowych w moich modelach. Może to w sumie być dlatego że im bardziej precyzyjny jest dataset w sensie pozycji piłkarza, tym mniej danych treningowych on zawiera, co powoduje gorsze przystosowanie się modelu do tych danych

6 Wnioski

Ogólnie wszystkie modele mają błąd bezwzględny w kilka milionów euro, a w niektórych przypadkach ten błąd może przekraczać kilkudziesięciu milionów euro. Może to być akceptujące dla najbogatszych klubów, ale dla średnich klubów które nie mają tak dużo pieniędzy i które poszukują piłkarzy którzy kosztują do kilku milionów euro(lub maksymalnie kilku dziesięciu milionów euro) te modele nie bardzo podchodzą dla oceniania wartości transferowej piłkarzy.

Prawdopodobnie można byłoby polepszyć nasze modele mając większą ilość danych, na przykład całą historię wystąpień piłkarza, jego popularność, ilość zdobytych trofeów, popularność ligi, klubu i tak dalej.