Julita Ołtusek
Łukasz Pik

Klasyfikacja piłkarzy pod względem skuteczności na zbiorze danych FIFA 2019 Complete Player Dataset (Kaggle)

Dokumentacja wstępna

1. Interpretacja tematu projektu

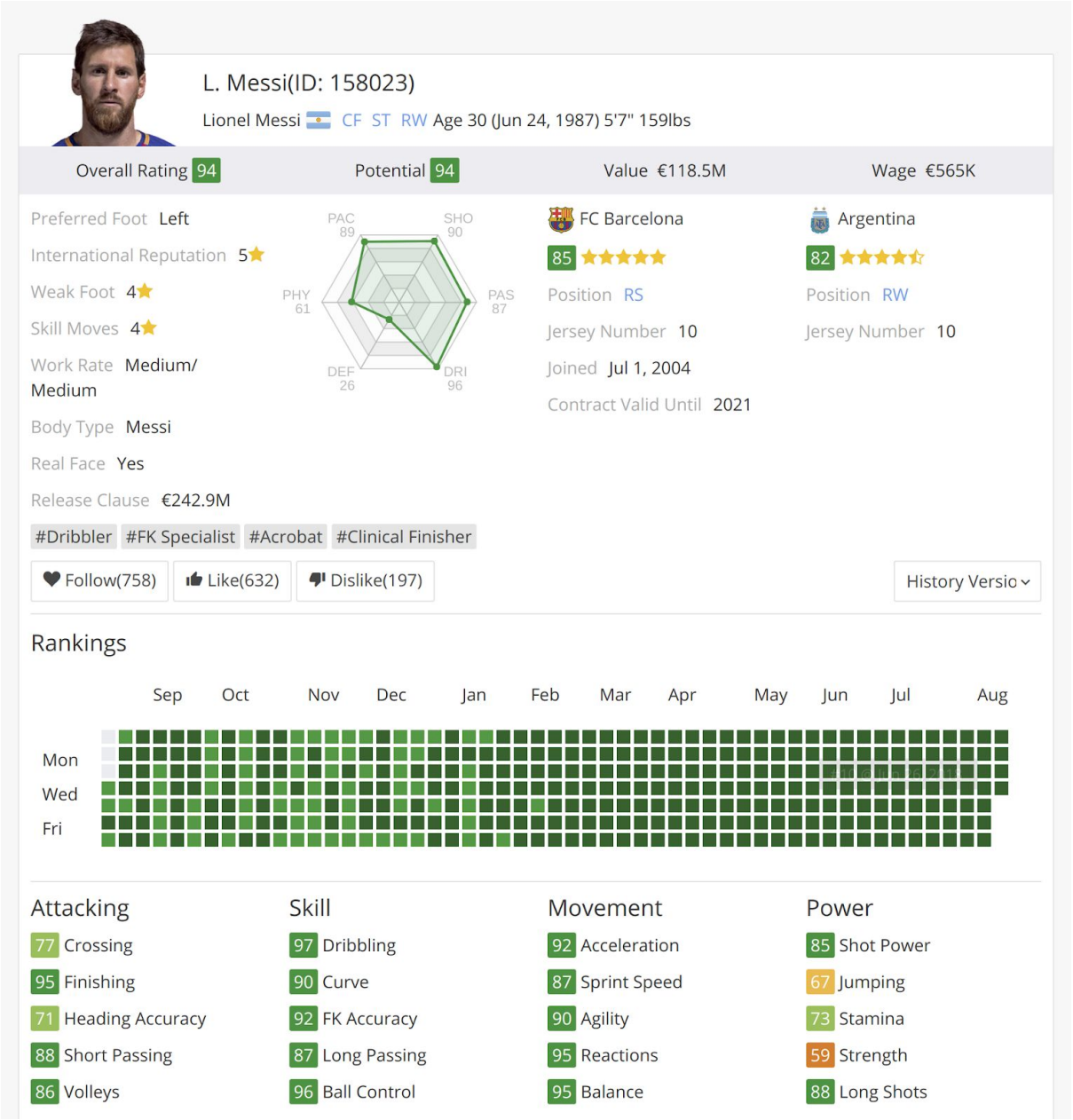
Zagadnieniem przeprowadzanej klasyfikacji jest predykcja skuteczności piłkarzy za pomocą wybranych metod uczenia maszynowego pod nadzorem. Główny cel projektu to budowa modeli klasyfikacyjnych oraz ich analiza na podstawie przeprowadzonych testów, z wykorzystaniem szeregu algorytmów, których implementacje są dostępne w pakietach języka R. Przeprowadzona analiza posłuży do oceny jakości klasyfikacji przez wytrenowane modele oraz ich porównanie. Źródłem danych użytym w projekcie będzie baza danych piłkarzy FIFA 2019.

2. Wybór atrybutu reprezentującego pojęcie docelowe

Zbiór danych postanowiliśmy podzielić na klasy względem atrybutu ogólnej oceny (overall). Atrybut ten przyjmuje wartości całkowite 0-100. Zdecydowaliśmy się podzielić ten zakres na 5 przedziałów w taki sposób, aby zbiór danych był zbalansowany, a więc dobierając takie wartości granic przedziałów, aby liczba elementów należących do każdej klasy była zbliżona. Skoro chcemy uzyskać 5 przedziałów, to wartościami granicznymi będą kolejne kwintyle rozkładu tej zmiennej. Każdej z grup przyporządkujemy etykietę, która będzie odpowiadać pozycji na wyznaczonej przez przedział skali (1-5). Na podstawie pozostałych cech, odpowiednio przygotowanych, przeprowadzona zostanie klasyfikacja piłkarzy pod względem skuteczności. Jeśli okaże się, że modele przy tak postawionym problemie nie mogą osiągnąć zbyt dobrej jakości, ponieważ klasy są zbyt do siebie podobne, wykonamy przekształcenie zadania na klasyfikację binarną. Rozróżnione klasy w takim przypadku to: 20% najlepszych piłkarzy oraz pozostałe 80% gorszych piłkarzy.

3. Opis danych

Zbiór danych wykorzystany w tym zadaniu pochodzi z FIFA 19 complete player dataset. Dane zostały przygotowane w oparciu o serwis https://sofifa.com/, z którego za pomocą skryptu indeksującego, zostały wydobyte dane o piłkarzach znajdujących się w grze FIFA 2019. Do danych należą między innymi: wzrost, wiek, statystyki ofensywne jak i defensywne, czy też refleks. Zbiór danych zawiera 18 207 rekordów, z których każdy jest opisany za pomocą 89 atrybutów. Są to zarówno atrybuty numeryczne (np. drybling, kondycja, wytrzymałość), jak i dyskretne (np. preferowana noga, pozycja). Na poniższej grafice widoczny jest widok szczegółowy dla jednego piłkarza, czyli pojedynczej próbki z naszego zbioru danych.



4. Przygotowanie danych wejściowych

Zadany zbiór danych został przygotowany w postaci pliku csv, co umożliwia jego prostą konwersję do postaci tabelarycznej. Jest już częściowo przetworzony i nie zawiera żadnych brakujących wartości. Niektóre atrybuty, takie jak logo klubu czy długość kontraktu, nie są przydatne w naszym zadaniu klasyfikacji i zostaną usunięte ze wstępnego zbioru danych. Takie cechy jak wzrost, które zostały podane w formacie tekstowym, ponieważ są zapisane w jednostkach imperialnych, zostaną zakodowane w formie wartości liczbowych, aby wydobyć z nich relację porządku. Większość cech liczbowych jest mierzonych w skali 0-100, dlatego standaryzacja prawdopodobnie nie będzie konieczna, ale tę decyzję zostawiamy na późniejszy etap, kiedy dokładniej zbadamy rozkłady cech.

5. Selekcja atrybutów

Z uwagi na dużą liczbę atrybutów, jaki dany zbiór posiada, tylko części z nich użyjemy do klasyfikacji. Dzięki temu ograniczymy złożoność obliczeniową, a także pozbędziemy się atrybutów, które nie są skorelowane z klasami. Aby zmaksymalizować jakość predykcji należy pozbyć się informacji zbędnych lub powtarzających się. W tym celu należy dowiedzieć się, jak wiele informacji o klasie niesie ze sobą dana cecha. Przy odrzuceniu części atrybutów można posłużyć się różnymi technikami wyboru cech. Mogą one korzystać z entropii, Indeksu Giniego lub w inny sposób obliczać korelację zmiennych. W naszym przypadku wykorzystamy funkcję *varimp* z pakietu *caret* środowiska R, która korzysta z różnych metod w zależności od algorytmu klasyfikacji, do jakiego jest zastosowana. Wybierzemy takie cechy, które zostaną wybrane jako ważne dla algorytmu las losowy.

6. Algorytmy klasyfikacji

6.1. Naiwny klasyfikator Bayesa

Jest to model cech niezależnych, który korzystając z twierdzenia Bayesa wyprowadza model prawdopodobieństwa. Ta metoda bardzo dobrze sprawuje się w przypadkach dużej liczby cech obiektów oraz jest stosunkowo łatwa do wykorzystania, a także bardzo skutecznie klasyfikuje, o ile założenie o wzajemnej niezależności atrybutów jest spełnione w znacznym stopniu. W projekcie zostanie wykorzystany klasyfikator z pakietu e1071.

6.2. Drzewo decyzyjne

Drzewa służą do analizy oraz klasyfikacji danych, na podstawie serii warunków znajdujących się w węzłach. Klasyfikacja polega na przejściu drzewa od korzenia do liścia, który daje nam jednoznaczną odpowiedź. Proces klasyfikacji z wykorzystaniem drzew decyzyjnych jest efektywny obliczeniowo, wyznaczenie kategorii przykładu wymaga w najgorszym razie przetestowania raz wszystkich jego atrybutów. Wykorzystana zostanie implementacja z pakietu rpart.
Parametry do strojenia:

- minsplit - minimalna liczba obserwacji, jaka musi być wykryta w węźle, aby był on rozważany do analizy,
- minbucket - minimalna liczba obserwacji, jaką musi posiadać każdy liść drzewa,
- cp - parametr odpowiedzialny za obcinanie podziałów małoznaczących. Każdy podział, który nie będzie przypasowania o co najmniej wartość parametru cp, zostanie pominięty,
- maxcompete - liczba wybieranych podziałów jakie drzewo ma posiadać,
- maxsurrogate - maksymalna liczba zastępczych podziałów,
- usesurrogate - parametr mówiący jak mają wyglądać podziały zastępcze
  dla wartości 0 - jeśli nie ma wartości podziału dla danej zasady, objekt nie jest dalej analizowany
  1 - używa zastępczych wartości, aby dokonać podziału na zasadzie, jeśli brakuje wszystkich zastępców, objekt nie jest analizowany
  2 - jeśli brakuje wszystkich zastępców, wtedy wykorzystuje ścieżkę większościową,
- xval - liczba walidacji krzyżowych,
- surrogatestyle - parametr kontrolujący wybór najlepszych zastępców. ,

- maxdepth - maksymalna głębokość finalnego drzewa.

6.3. Las losowy

Las losowy jest zbudowany w wielu klasyfikatorów, którymi są właśnie drzewa decyzyjne. Dane przechodzą przez każde z drzew, a następnie gdy klasyfikacja zostanie przez nie zakończona, wyniki są agregowane i wybierany jest poprzez głosowanie ostateczny wynik klasyfikacji. Do lasu losowego wykorzystana zostanie implementacja z pakietu randomForest.

Parametry do strojenia:

- nodesize - minimalny rozmiar liści (liczba próbek ze zbioru trenującego),
- maxnodes - maksymalna liczba liści,
- ntree - liczba drzew,
- replace - parametr mówiący, czy powinny być używane zamienniki
- mtry -liczba zmiennych losowo wybranych na kandydatów na każdym podziale. Domyślnie jest to wartość sqrt(p), gdzie p to liczba zmiennych w analizowanych danych.
- cutoff - wektor proporcji głosów, używany do obcinania
- strata - zmienna używana do losowania warstwowego
- sampsize - wielkość przykładów do rysowania

7. Testowanie i ocena jakości modeli

Kiedy model będzie już zbudowany, można będzie przystąpić do oceny jego jakości. Ten etap polega na sprawdzeniu na zbiorze testowym, jaka jest skuteczność predykcji wytrenowanego modelu. Przy ocenie klasyfikatora można wziąć pod uwagę wiele kryteriów. Opisujemy dalej, jakie miary jakości oraz procedury oceny zastosujemy.

7.1. Macierz pomyłek

Przy ocenie jakości modeli bardzo pomocna jest macierz pomyłek nazywana również macierzą błędów (ang. error matrix) lub konfuzji (ang. confusion matrix) . Forma tabeli pozwala na przejrzystą wizualizację wyników testów klasyfikatora. To macierz kwadratowa o wymiarach *nxn*, gdzie *n* to liczba klas. Każdy wiersz reprezentuje klasy faktyczne, natomiast każda kolumna – klasy przewidziane. Zatem w komórce *mij* zapisuje się liczbę przypadków testowych należących do klasy *yi*, zaklasyfikowanych do klasy *yj*. Na głównej przekątnej znajdują się przypadki zaklasyfikowane poprawnie. Macierz jest wykorzystywana do wyznaczania miar jakości klasyfikatora.

cała próba		klasy przewidziane				
		klasa 1	klasa 2	…	klasa n	Σ
klasy faktyczne	klasa 1	m_{11}	m12	…	m_{1n}	$\sum_{j=1}^{n} m_{1j}$
	klasa 2	m_{21}	m_{22}	…	m_{2n}	$\sum_{j=1}^{n} m_{2j}$
	…	…	…	…	…	…
	klasa n	m_{n1}	m_{n2}	…	m_{nn}	$\sum_{j=1}^{n} m_{nj}$
	Σ	$\sum_{i=1}^{n} m_{i1}$	$\sum_{i=1}^{n} m_{i2}$	…	$\sum_{i=1}^{n} m_{in}$	$\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}$

Tabela 1. Przykładowa macierz pomyłek dla klasyfikacji wieloklasowej.

7.2. Miary jakości

W projekcie użyjemy 3 miar jakości klasyfikacji, które są jednymi z najpopularniejszych, a także dodatkowej metody oceny - krzywej ROC. W przypadku, kiedy zdecydujemy się na wariant wieloklasowy, w celu oceny modeli sprowadzimy je do zadania binarnego pod postacią OVA (one vs all - jedna kontra pozostałe). W dalszej części tego punktu użyliśmy następujących oznaczeń:
X - wektor atrybutów
k - numer klasy
K - liczba wszystkich klas
Y - etykieta klasy
d - klasyfikator
m_{ij} - wartość z pola macierzy konfuzji

Dokładność (ang. accuracy)

Dokładność to jedna z popularnych miar ocen klasyfikatorów. Jest oznaczana przez procentową część przykładów testowych poprawnie zaklasyfikowanych do odpowiadających im klas. W taki sposób określa się prawdopodobieństwo, że nowy przykład zostanie poprawnie zaklasyfikowany.

$$accuracy(d) = P(d(X) = Y) = \frac{\sum_i m_{ii}}{\sum_{i,j} m_{ij}}$$

Precyzja (ang. precision)

Precyzja to miara, która określa prawdopodobieństwo tego, że obiekt faktycznie należy do danej klasy, jeśli został do niej zaklasyfikowany. Jest to stosunek liczby obiektów poprawnie zaklasyfikowanych do danej klasy do wszystkich obiektów, które zostały zaklasyfikowane jako należące do niej (łącznie z błędnie zaklasyfikowanymi). Precyzja liczona jest oddzielnie dla każdej klasy.

$$precision_k(d) = P(Y = y_k | d(X) = y_k) = \frac{m_{kk}}{\sum_j m_{jk}}$$

Czułość (ang. recall, sensitivity)

Czułość określa stosunek liczby poprawnie zaklasyfikowanych obiektów z danej klasy do wszystkich obiektów, które rzeczywiście do niej należą. Innymi słowy jest to prawdopodobieństwo poprawnej klasyfikacji do danej klasy pod warunkiem, że obiekt rzeczywiście do niej należy. Czułość podobnie jak precyzja liczona jest oddzielnie dla każdej klasy.

$$recall_k(d) = P(d(X) = y_k | Y = y_k) = \frac{m_{kk}}{\sum_j m_{kj}}$$

Krzywa ROC (ang. Receiver Operating Characteristic curve)

Krzywa ROC to graficzna metoda oceny jakości klasyfikacji. W układzie współrzędnych (FP rate, TP rate) przedstawiana jest wizualizacja punktów pracy modeli klasyfikacji. Pole pod krzywą ROC nazywa się AUC (Area Under the Curve) i jego zakres wartości wynosi [0;1]. Im bliżej 1, tym model lepiej radzi sobie z klasyfikacją. Dla przypadku wieloklasowego sporządzimy charakterystyki dla każdego wariantu OVA bądź jedną wspólną, po uśrednieniu wartości z komórek poszczególnych macierzy pomyłek.

7.3. Procedury oceny jakości

Zbiór testowy jest zwykle częścią populacji danych, pozostałą po oddzieleniu wcześniej zbioru trenującego. Dzięki temu, że trenowanie modelu przebiega na innych danych, niż jego testowanie, można sprawdzić, jak model zachowuje się w przypadku nowych dla niego i zupełnie nieznanych danych. Pomaga to zapobiec przeuczeniu (ang. overfitting), czyli nadmiernemu dopasowaniu danych do zestawu treningowego (czasem też pośrednio testowego). Klasyfikator przewiduje klasy dla przypadków testowych, a ich wartości porównywane są z rzeczywistymi znanymi klasami, po czym wyznaczane są miary jakości opisane w poprzednim punkcie.

7.3.1. Walidacja krzyżowa

Do badania skuteczności klasyfikatorów często wykorzystuje się metodę znaną w statystyce jako walidacja krzyżowa (ang. cross-validation). Procedura w ogólnym przypadku przebiega w taki sposób, że zarówno budowa klasyfikatora, jak i jego testowanie wykonywane są wielokrotnie, za każdym razem przy innym podziale danych na zbiór treningowy i testowy. Następnie wyniki wszystkich iteracji są łączone (np. uśredniane), co pozwala oszacować skuteczność predykcji modelu. Rozróżnia się kilka rodzajów tej metody, jednak my zdecydowaliśmy się wybrać **K-krotną walidację ze zbiorem testującym i weryfikującym**. Liczność przykładów w zbiorze danych wynosi ponad 18 tysięcy, dlatego uznaliśmy, że możemy sobie pozwolić na wydzielenie zbioru weryfikującego. Wartość liczby podziałów k ustalimy w trakcie eksperymentów, aby na ile to możliwe, zrównoważyć wariancję i obciążenie estymacji.

7.3.2. Makro- i mikro-uśrednianie

Po przeprowadzeniu szeregu testów przy użyciu walidacji krzyżowej i wyznaczeniu wartości miar jakości dla poszczególnych klas (precyzja i czułość), można je uśrednić, aby uzyskać

ogólne rezultaty klasyfikacji. W tym celu wykorzystuje się zwykle jedną z dwóch metod: makro- i mikro- uśredniania.

Makro-uśrednianie
W metodzie makro-uśredniania obliczona średnia miara to po prostu średnia miar dla poszczególnych klas.

$$\text{measure}_{\text{macro}}(d) = \frac{1}{\text{K}}\sum_{\text{k=1}}^{\text{K}} \text{meaure}_{\text{k}}(\text{d})$$

Ważnym aspektem jest tutaj brak wrażliwości na różnice w licznościach zbiorów należących bądź zaklasyfikowanych do różnych klas, gdyż każda klasa otrzymuje jednakową wagę.

Mikro-uśrednianie
W metodzie mikro-uśredniania średnia miara jest liczona w nieco bardziej skomplikowany sposób. Ogólnie mówiąc, w każdym miejscu gdzie występują wartości dla konkretnej klasy, trzeba je zsumować po wszystkich klasach. Wzór poniżej odpowiada zarówno średniej precyzji, jak i czułości, a także dokładności, gdyż wartości w mianowniku, czyli suma próbek należących do wszystkich klas i suma próbek sklasyfikowanych są równe, w obu przypadkach to liczność całego zbioru testowego.

$$precision_{micro}(d) = recall_{micro}(d) = \text{accuracy(d)} = \frac{\sum_{k=1}^{K} m_{kk}}{\sum_{i,j} m_{ij}}$$

W tym przypadku, jeśli zbiór danych nie jest zbalansowany, uśredniona miara uwzględnia bardziej te liczniejsze klasy. Np. dla klasyfikacji binarnej, gdy do jednej klasy należy 10% wszystkich próbek, a do drugiej 90%, może się zdarzyć że uśredniona miara będzie wskazywać wysoką wartość, nawet jeśli większość przykładów reprezentujących mniej liczną klasę będzie klasyfikowana nieprawidłowo.

Ponieważ założyliśmy na początku, że próbki danych podzielimy na klasy w taki sposób, aby uzyskać zbalansowany zbiór, nie ma właściwie znaczenia, którą z metod uśredniania wybierzemy, bo każda z nich powinna dawać zbliżone wyniki. Do wariantu wieloklasowego postanowiliśmy zatem zastosować makro-uśrednianie. W przypadku ewentualnej klasyfikacji binarnej, z powodu niezbalansowanego zbioru danych (stosunek liczebności w klasach wynosił będzie około 1:4) uznaliśmy, że lepiej będzie posłużyć się mikro-uśrednianiem, aby nie zniwelować oceny modelu dla klasy mniej licznej (w naszym przypadku 20% najbardziej skutecznych piłkarzy).

8. Podsumowanie

Metody klasyfikacji, które wymieniliśmy zostaną poddane testom, podczas których dopasujemy możliwe do strojenia parametry. Będzie to miało na celu uzyskanie jak najwyższej jakości klasyfikacji, przy jednoczesnych staraniach uniknięcia nadmiernego dopasowania modeli do danych (zarówno bezpośredniego do danych trenujących, jak i

pośredniego do testowych). Następnie zostaną porównane wyniki zbudowanych modeli, uzyskane na zbiorze walidacyjnym.