Sprawozdanie 3 z laboratorium: Uczenie maszynowe

Studium przypadku - analiza zbioru FIFA 18

27 czerwca 2019

Prowadzący: dr hab. inż. Maciej Komosiński

Autor: **Michał Kałczyński** inf127317 ISWD michal.kalczynski@student.put.poznan.pl

Zajęcia środowe, 16:50.

Oświadczam, że niniejsze sprawozdanie zostało przygotowane wyłącznie przez powyższego autora, a wszystkie elementy pochodzące z innych źródeł zostały odpowiednio zaznaczone i są cytowane w bibliografii.

1 Wstęp

Analizowany przeze mnie zbiór pochodzi z serwisu Kaggle[1]. Zbiór ten zawiera podstawowe informacje na temat zawodników z gry Fifa 18. Dostępnych jest ponad 17 tysięcy zawodników, a każdy z nich opisany jest przez ponad 70 atrybutów, takich jak jego wartość rynkowa, pensja, wiek, czy posiadane umiejętności piłkarskie.

W ramach tego projektu zająłem się trzema różnymi problemami:

- Predykcja pozycji piłkarza
- Predykcja ogólnej oceny piłkarza
- Predykcja wartości rynkowej piłkarza

1.1 Preprocessing danych

Zbiór danych jest bardzo dobrej jakości i nie wymagał większego wstępnego przetworzenia. Jedynym problemem jaki napotkałem, było kilkukrotne wystąpienie sumy w atrybutach opisujących umiejętności piłkarskie. Atrybuty te powinny zawierać liczby całkowite z przedziału 0-100, lecz w kilku miejscach można było znaleźć wartości takie jak '65+4'. W takich wypadkach wyrzucałem dodatkowe wartości znajdujące się od znaku dodawania, by móc mieć jednolite dane liczbowe.

1.2 Wykorzystane metody uczenia maszynowego

W projekcie użyłem czterech metod uczenia maszynowego. Wykorzystałem ich implementację z biblioteki Scikit Learn w języku Python:

- Algorytm kNN - klasa sklearn.neighbors.KNeighborsClassifier
- Drzewo decyzyjne - klasa sklearn.tree.decisiontreeclassifier - użyty jest tu algorytm CART, który zbliżony jest do algorytmu C4.5
- Sieć neuronowa - sklearn.neural_network.MLPRegressor
- Klasyfikator ZeroR - sklearn.dummy.DummyClassifier(strategy='most_frequent')

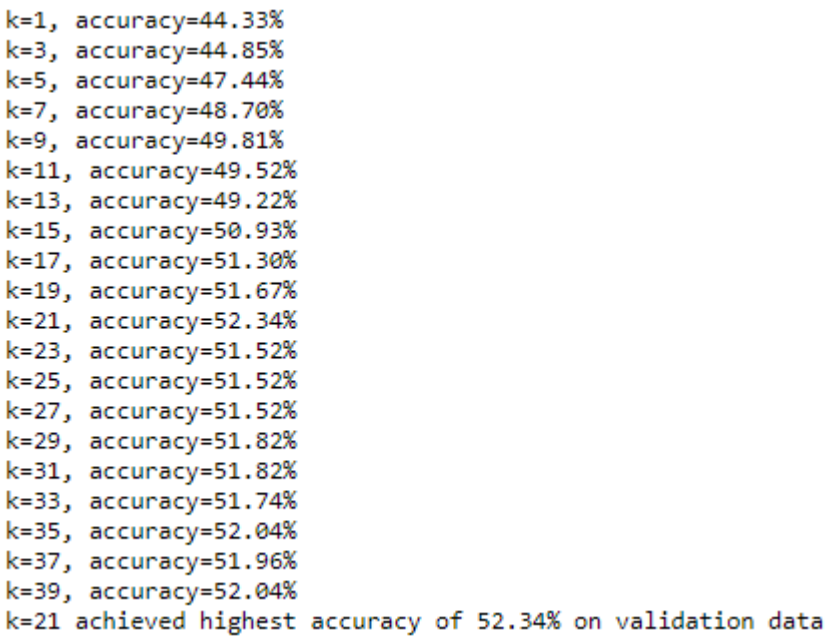
2 Predykcja pozycji piłkarza

W tym rozdziale postanowiłem rozpoznać pozycję (15 klas decyzyjnych), na której gra dany piłkarz, na podstawie tylko jego umiejętności piłkarskich, takich jak: przyspieszenie, kontrola piłki czy umiejętność wykończenia akcji. W zbiorze danych dostępna jest informacja o pozycjach na jakich piłkarz preferuje grać (wykorzystując ją będziemy mogli uczyć i testować program), ale dopuszcza, by jeden piłkarz miał więcej nominalnych pozycji. W takich przypadkach brałem pod uwagę, zarówno podczas uczenia jak i testowania, tylko pierwszą pozycję, jaka się w tym polu pojawiła.

2.1 Algorytm kNN

Zbiór danych podzieliłem na zbiór treningowy, walidacyjny i testowy w proporcjach odpowiednio 67.5%, 7.5% i 25%. Zbiór walidacyjny wykorzystałem do wyboru najlepszej wartości parametru k (21), co zostało pokazane na Rysunku 1.

[1] https://www.kaggle.com/thec03u5/fifa-18-demo-player-dataset



Rysunek 1: Algorytm kNN: przedstawienie wyników wyboru optymalnego parametru k, za pomocą zbioru walidującego

Algorytm kNN na zbiorze testowym uzyskał dokładność 50.8%. Dokładniejsze wyniki przedstawiam na Rysunku 2. Użyte oznaczenia są angielskimi skrótami nazw poszczególnych pozycji, np. CAM (ang. Centre Attacking Midfield) oznacza ofensywnego środkowego pomocnika, a CB (ang. Centre-back) środkowego obrońcę. Widać, że pozycja bramkarza (ang. GK) była zawsze dobrze przewidziana, ponieważ odznaczają się oni umiejętnościami, bardzo wyraźnie różniącymi się od umiejętności piłkarzy na innych pozycjach.

2.2 Drzewo decyzyjne

W celu wykorzystania drzewa do klasyfikacji zbiór danych podzieliłem na zbiór uczący i testowy w proporcjach odpowiednio 75% i 25%. Na zbiorze testowym otrzymałem dokładność 44.86%.

Na Rysunku 3 przedstawiam macierz pomyłek, ilustrującą działanie drzewa. Tutaj także widać, że nie ma problemów z klasyfikacją bramkarzy. Można dostrzec także sytuacje, że piłkarze są klasyfikowani często na odpowiednie im pozycje symetryczne na boisku, np. LM (lewy pomocnik) prawie tak samo często klasyfikowany jest na pozycję RM (prawy pomocnik), ponieważ gracze ci mają w praktyce podobne umiejętności i mogą grać na obu tych pozycjach lub zależy to od ich preferowanej nogi. Niestety w dostępnym zbiorze nie ma informacji o tym, czy piłkarz jest prawo czy lewonożny.

2.3 Klasyfikator ZeroR

W celu lepszego zobrazowania i porównania osiągniętych w poprzednich paragrafach wyników, postanowiłem także przedstawić wyniki dla klasyfikatora ZeroR. Podobnie jak w poprzednim paragrafie, także i teraz zbiór danych podzieliłem na zbiór uczący i testowy w

	precision	recall	f1-score	support
CAM	0.23	0.21	0.22	274
CB	0.71	0.84	0.77	680
CDM	0.33	0.29	0.31	352
CF	0.00	0.00	0.00	41
CM	0.32	0.52	0.40	450
GK	1.00	1.00	1.00	525
LB	0.37	0.40	0.38	376
LM	0.24	0.15	0.18	370
LW	0.12	0.03	0.05	100
LWB	0.00	0.00	0.00	34
RB	0.32	0.26	0.29	311
RM	0.20	0.17	0.18	314
RW	0.05	0.01	0.02	95
RWB	0.00	0.00	0.00	26
ST	0.66	0.83	0.73	548
accuracy			0.51	4496
macro avg	0.30	0.31	0.30	4496
weighted avg	0.47	0.51	0.48	4496

Rysunek 2: Algorytm kNN: wynik działania dla k=21

proporcjach odpowiednio 75% i 25%. Klasyfikator ZeroR uzyskał dokładność 14.75% przewidując za każdym razem pozycję środkowego obrońcy (skrót CB).

2.4 Wnioski

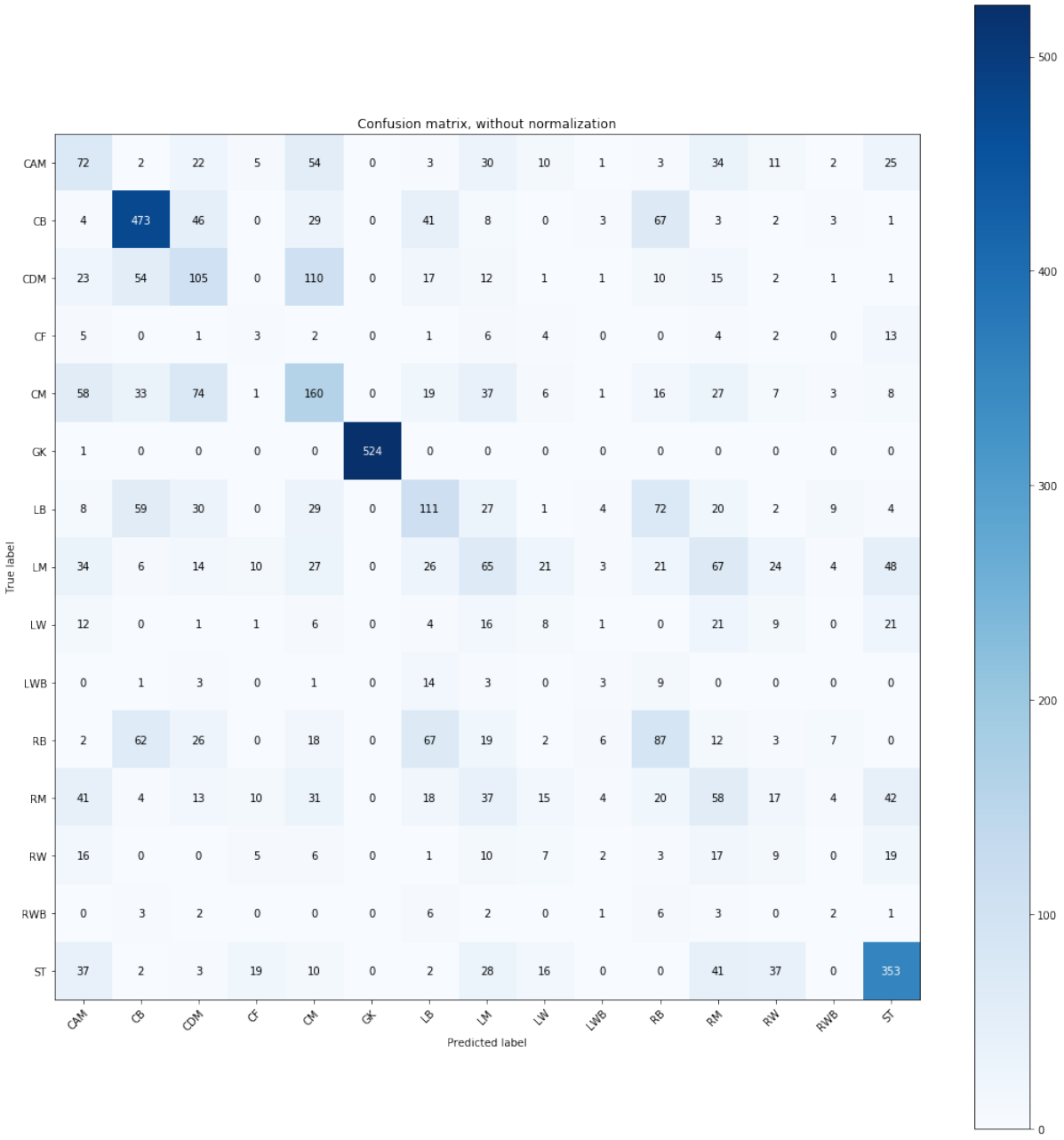
Otrzymane wyniki zamieszczam w poniższej tabeli

kNN	drzewo	zeroR
50.8%	44.86%	14.75%

Lepszy wynik w tym problemie dał algorytm kNN niż drzewo decyzyjne. Dokładność rzędu 50% nie wydaje się być wysoka, ale trzeba wziąć pod uwagę fakt, iż zarówno podczas nauki jak i testowania braliśmy pod uwagę tylko jedną możliwą klasę decyzyjną (w oryginalnym zbiorze często występowało ich więcej). Mogło to mieć istotny wpływ na ostateczną dokładność predykcji, ponieważ wielu piłkarzy ma odpowiednie umiejętności, by grać na różnych pozycjach, szczególnie gdy wyróżnia się ich aż 15. Dodatkowym utrudnieniem jest fakt, że zbiór nie ma informacji o tym czy piłkarz jest prawo czy lewo nożny (lub jak dobrze posługuje się gorszą nogą), co także ma istotny wpływ na jakiej pozycji gra, np. lewa lub prawa obrona.

3 Predykcja ogólnej oceny piłkarza

Każdy zawodnik ma atrybut dotyczący ogólnej jego oceny. Może on przyjmować wartości całkowite od 0 do 100, lecz w zbiorze znajdują się tylko zawodnicy mający ocenę od 46 do 94. Aby móc sprowadzić problem do problemu klasyfikacji, w którym byłoby mniej klas decyzyjnych, postanowiłem na podstawie tego atrybutu stworzyć nowy. Atrybut ten został stworzony w następujący sposób:



Rysunek 3: Drzewo decyzyjne: macierz pomyłek

```
k=1, accuracy=81.25%
k=3, accuracy=85.32%
k=5, accuracy=86.14%
k=7, accuracy=86.95%
k=9, accuracy=87.32%
k=11, accuracy=87.92%
k=13, accuracy=87.84%
k=15, accuracy=88.21%
k=17, accuracy=88.21%
k=19, accuracy=88.07%
k=21, accuracy=88.29%
k=23, accuracy=88.44%
k=25, accuracy=88.58%
k=27, accuracy=88.29%
k=29, accuracy=88.21%
k=31, accuracy=87.69%
k=33, accuracy=87.47%
k=35, accuracy=87.25%
k=37, accuracy=86.95%
k=39, accuracy=87.47%
k=25 achieved highest accuracy of 88.58% on validation data
```

Rysunek 4: Algorytm kNN: przedstawienie wyników wyboru optymalnego parametru k, za pomocą zbioru walidującego

Ogólna ocena	Wartość nowego atrybutu	Ilość piłkarzy
<46, 60)	Słaby	2934
<60, 70)	Przeciętny	9290
<70, 80)	Dobry	5238
<80, 90)	Gwiazda	508
<90, 94>	Legenda	11

Tak stworzony atrybut został atrybutem decyzyjnym dla rozważanego problemu.

3.1 Algorytm kNN

Podobnie jak w poprzednim rozdziale tak i tutaj podzieliłem zbiór danych na zbiory uczący (67.5%), walidujący (7.5%) oraz testowy (25%). Po czym przy pomocy zbioru walidującego wybrałem parametr k=25 (Rysunek 4).

Na zbiorze testowym otrzymałem dokładność 86.74%. Na Rysunku 5 widać, że wszyscy 4 piłkarze, którzy są legendami zostali wytypowani jako gwiazdy. Jest to z pewnością związane z tym, że zawodników tych jest w całym zbiorze bardzo mało (11), a dobrane k wynosiło aż 25. Dla klasy gwiazda widać też niską wartość recall (0.27), co oznacza, że sporo graczy słabszych musiało zostać niesłusznie zaklasyfikowanych właśnie jako gwiazdy.

	precision	recall	f1-score	support
Dobry	0.86	0.82	0.84	1279
Gwiazda	0.88	0.27	0.41	133
Legenda	0.00	0.00	0.00	4
Przeciętny	0.86	0.94	0.90	2363
Słaby	0.90	0.82	0.86	717
accuracy			0.87	4496
macro avg	0.70	0.57	0.60	4496
weighted avg	0.87	0.87	0.86	4496

Rysunek 5: Algorytm kNN: wynik działania dla k=21

3.2 Drzewo decyzyjne

Zbiór danych podzieliłem na zbiór uczący i testowy w proporcjach odpowiednio 75% i 25%. Na zbiorze testowym otrzymałem dokładność 84.12%.

Na Rysunku 6 przedstawiam macierz pomyłek, ilustrującą działanie drzewa. Jak widać drzewo także nie potrafiło poprawnie zaklasyfikować piłkarzy jako legend, wszystkich ich zaklasyfikowało jako gwiazdy.

3.3 Klasyfikator ZeroR

Także w tym przypadku postanowiłem porównać wyniki względem klasyfikatora ZeroR. Zbiór został podzielony na zbiór uczący i testowy w proporcjach odpowiednio 75% i 25%.

Klasyfikator ZeroR uzyskał dokładność 51% przewidując za każdym razem, że gracz jest przeciętny.

3.4 Wnioski

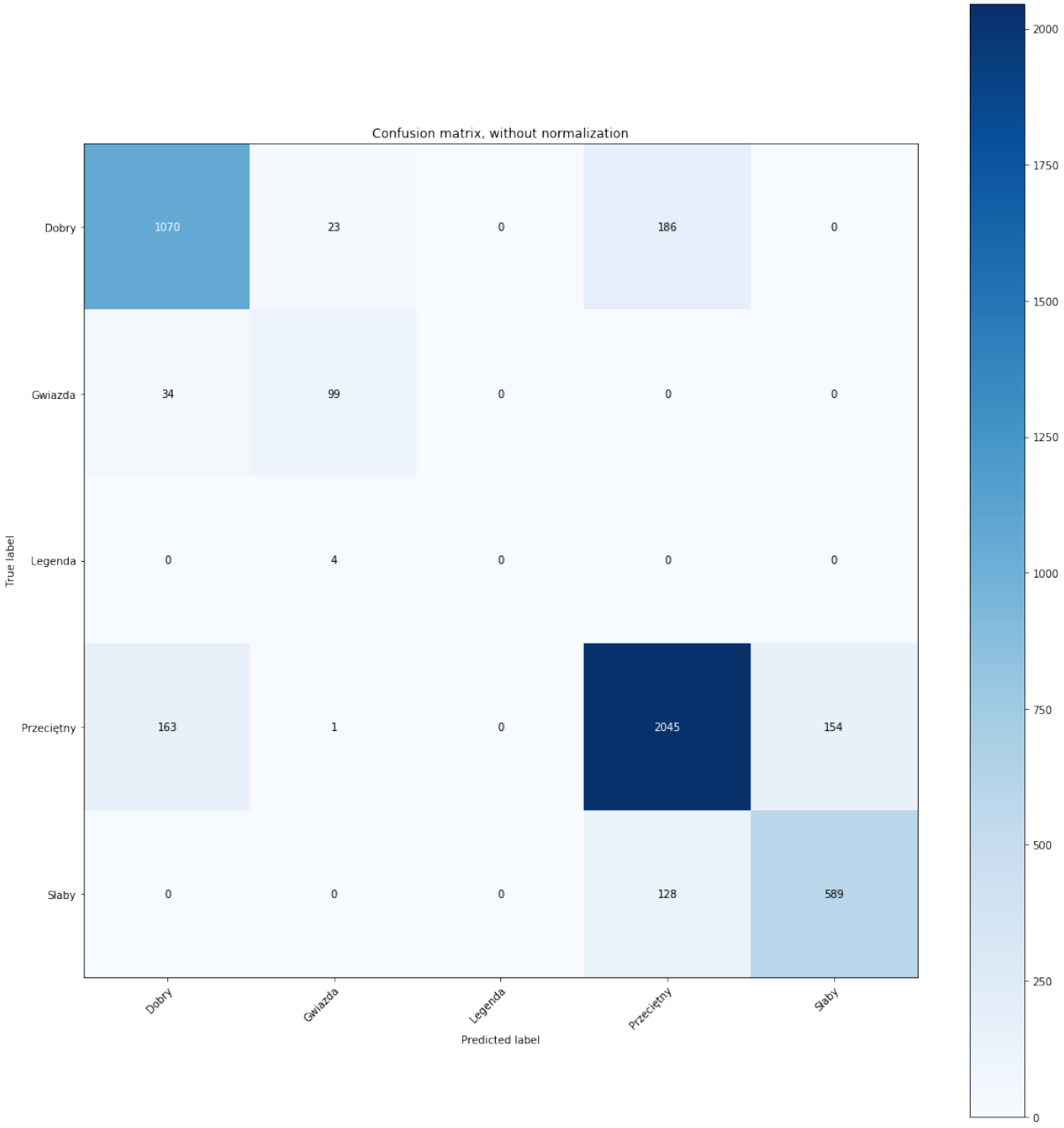
Otrzymane wyniki zostały zebrane w poniższej tabeli

kNN	drzewo	zeroR
86.74.8%	84.12%	51%

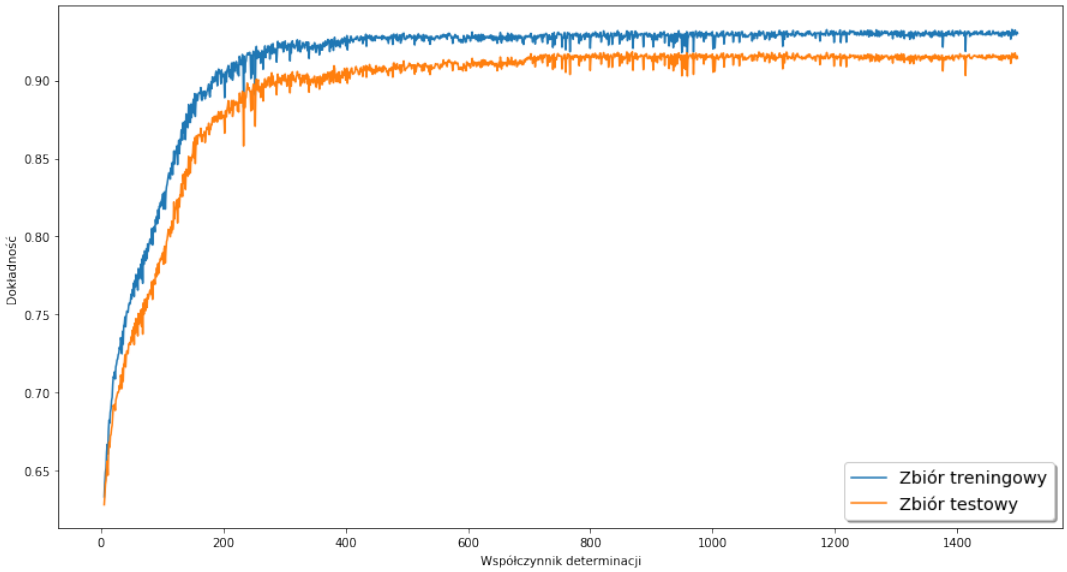
Także w tym problemie minimalnie lepszym rozwiązaniem było użycie algorytmu kNN. Oba algorytmy nie dały jednak rady poprawnie zaklasyfikować piłkarzy legend, ponieważ było ich tylko jedenastu. Być może lepszy wynik względem tych piłkarzy, dałoby się uzyskać wybierając mniejsze k w algorytmie kNN. Trzeba by jednocześnie spodziewać się nieznacznego spadku ogólnej dokładności klasyfikowania. Innym sposobem mogłaby być ingerencja w zbiór danych i stworzenie dodatkowych zawodników, którzy według nas klasyfikują się na miano legend.

4 Predykcja wartości rynkowej piłkarza

Jednym z dostępnych w zbiorze atrybutów jest wartość rynkowa piłkarza. W tym rozdziale zamierzam przewidzieć ją, biorąc pod uwagę atrybuty związane z umiejętnościami gracza, jego wiekiem, ogólną oceną, przewidywanym potencjałem rozwoju.



Rysunek 6: Drzewo decyzyjne: macierz pomyłek



Rysunek 7: Wykres dokładności dla zbioru uczącego i testowego podczas uczenia się sieci

Wartość rynkowa piłkarza w zbiorze przedstawiona była jako napis, który zawierał jednostkę monetarna (znak euro) oraz mógł mieć skrót 'M' lub 'K', oznaczające odpowiednio milion lub tysiąc. Z tego powodu musiałem pozbyć się tych dodatkowych znaków i przeliczyć wartość do wspólnej jednostki. Dodatkowo pewna ilość piłkarzy miała wartość równą 0. Postanowiłem więc nie brać pod uwagę zawodników, którzy mieli wartość rynkową poniżej 1000 euro, aby mieć pewność, że piłkarze ci nie wpłyną negatywnie na działanie sieci. Nie było ich wielu, dokładnie 256 na ponad 17 tysięcy zawodników.

Z uwagi na dużą rozpiętość wartości piłkarzy (od kilku tysięcy do kilkuset milionów) postanowiłem znormalizować te wartości, poprzez standaryzację.

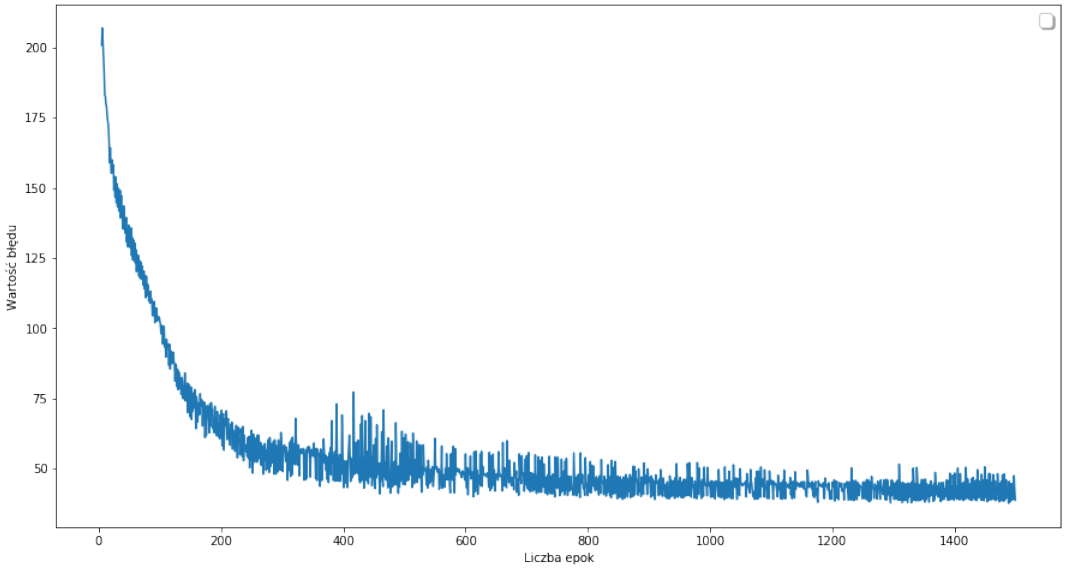
4.1 Sieć neuronowa

Do predykcji wartości wykorzystałem sieć neuronową typu feed-forward z czterema warstwami ukrytymi po 40 neuronów każda. Sieć była uczona przez 1500 iteracji, wykorzystując funkcję aktywacji ReLU, solver Adam oraz początkową prędkość uczenia 0.003. Zbiór danych został podzielony na zbiór uczący i testowy w proporcjach 3:1.

Wynikiem dopasowania modelu do danych jest tutaj współczynnik determinacji R^2. Dla zbioru uczącego wyniósł on 0.9302, a dla testowego 0.9145.

Na Rysunku 7 przedstawiona jest wartość współczynnika determinacji dla zbioru uczącego i testowego podczas nauki sieci. Rysunek 8 pokazuje wartość funkcji straty podczas nauki.

Przykładowe wyniki dla 5 czołowych piłkarzy w grze (mogli oni także trafić do zbioru uczącego)



Rysunek 8: Wartość funkcji straty podczas uczenia się sieci

Nazwisko piłkarza	Wycena w grze [mln]	Wycena sieci [mln]
Ronaldo	95.5	105.58
Messi	105	105.68
Neymar	123	94.74
Suarez	97	89.36
Neuer	61	93.38
Lewandowski	92	92.32