Rozpoznawanie aktywności fizycznej za pomocą danych z telefonu

Raport z projektu

Małgorzata Wachulec, Mateusz Bąkała, Wojciech Bogucki

16 czerwca 2019

Spis treści

1 Wstęp

1.1 Źródło i charakterystyka danych

Zbiór danych wykorzystany w tym projekcie to Human Activity Recognition Using Smartphones Data Set, pochodzący z UCI - Machine Learning Repository. Dotyczy on eksperymentu przeprowadzonego na 30 ochotnikach w wieku od 19 do 48 lat, podczas którego każdy z nich wykonywał sześć codziennych aktywności, takich jak: chodzenie, wchodzenie po schodach, schodzenie ze schodów, stanie, siedzenie oraz leżenie. Podczas eksperymentu każdy z ochotników miał przymocowanego w pasie Samsunga Galaxy S2, który za pomocą wbudowanych akcelerometru oraz żyroskopu, mierzył trójosiowe przyspieszenie liniowe oraz trójosiową prędkość kątową. Zmienne znajdujące się w głównej ramce danych to znormalizowane do przedziału $[-1, 1]$ statystyki przekształconych sygnałów, m.in. średnia, odchylenie standardowe, korelacja, przedział międzykwartylowy, kurtoza, entropia czy kąt między wektorami sygnałów.

Ze zródła danych wiemy, że każdy eksperyment był nagrywany, a etykiety 1-6 odpowiadające czynnościom wykonywanym przez uczestników, były ręcznie przypisane. Dodatkowo oryginalny zbiór danych został podzielony na zbiór treningowy (70% losowo wybranych obserwacji) i zbiór testowy (pozostałe 30% obserwacji).

1.2 Cel projektu

Celem tego projektu jest przeanalizowanie wyżej opisanych danych i zastosowanie algorytmu klasteryzującego tak, aby otrzymać klastry odpowiadające czynnościom wykonywanym przez uczesników eksperymentu.

2 Eksploracyjna analiza danych

2.1 Dostępne dane

Oryginalny zbiór danych podzielony był na kilka plików, ponieważ autorzy chcieli go dostosować do zadań klasyfikacji wieloklasowej. My jednak zajmowaliśmy się klasteryzacją, dlatego musieliśmy dokonać paru drobnych przekształceń.

Kluczową czynnością było połączenie zbiorów treningowego oraz testowego. W rezultacie z plików `X_train.txt` oraz `X_test.txt` utworzyliśmy jedną ramkę danych `X` o wymiarach 10299 obserwacji na 561 zmiennych, natomiast z plików `y_train.txt` oraz `y_test.txt` utworzyliśmy wektor `y` wykorzystywany przez nas przy ocenie jakości klasteryzacji.

Wektor etykiet zawiera liczby całkowite z przedziału [1, 6], które opisują rodzaj czynności opisywany przez daną obserwację. Są to czynności:

- dynamiczne:
  - `WALKING` – chodzenie,
  - `WALKING_UPSTAIRS` – wchodzenie po schodach,
  - `WALKING_DOWNSTAIRS` – schodzenie ze schodów,
- statyczne:
  - `SITTING` – siedzenie,
  - `STANDING` – stanie,
  - `LAYING` – leżenie.

Przy użyciu biblioteki `DataExplorer` wyprodukowaliśmy krótki opis podstawowych cech otrzymanego zbioru `X`.

Tablica 1: Podstawowe statystyki zbioru `X`

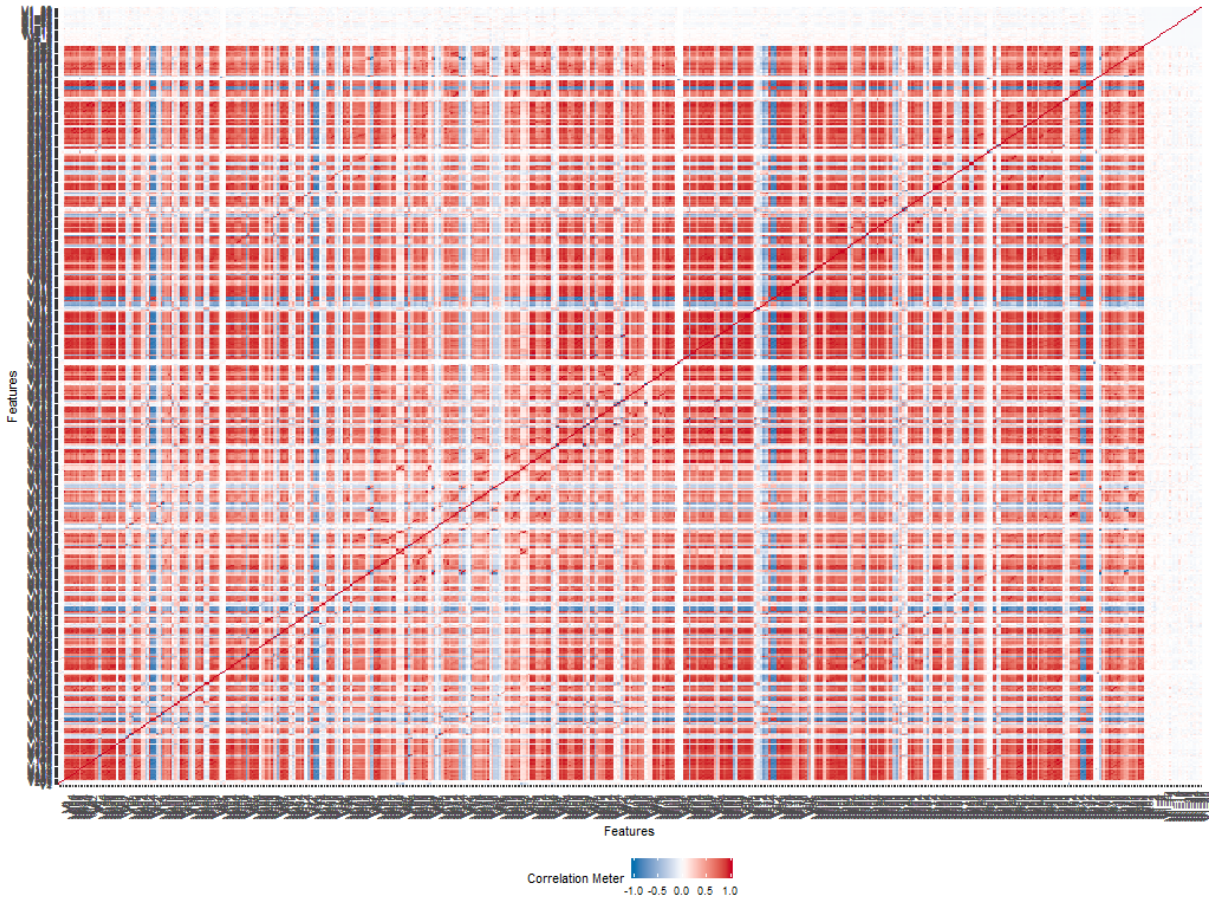
Statistic	Value
rows	10299
columns	561
discrete columns	0
continuous columns	561
all missing columns	0
total missing values	0
total observations	5777739
memory usage	46326104

2.2 Pozostałe dane

Dane przechowywane w zbiorze `X` to zestawy statystyk typu minimum, mediana czy kurtoza odnoszących się do różnych typów pomiarów. Trzeba jednak dodać, że wartości te zostały następnie zestandaryzowane i ograniczone do przedziału [-1, 1]. Wartości pomiarów w trzech osiach (X, Y i Z) znajdują się zaś w folderach `Inertial Signals`, jednak zdecydowaliśmy się korzystać tylko z danych zagregowanych. Wyniki pomiarów są szeregami czasowymi, które wymagają innego, specjalnego traktowania i niezbyt nadają się do typowego klastrowania.

2.3 Korelacje między zmiennymi

Z uwagi na dużą liczbę statystyk opisujących każdy typ pomiaru, spodziewaliśmy się sporej liczby skorelowanych zmiennych. W rzeczy samej, poniższa (1) grafika potwierdziła nasze obawy.



Rysunek 1: Macierz korelacji dla wyjściowego zbioru danych

Trochę zaskakujące było jednak, że również statystyki różnych typów pomiarów były w dużej mierze mocno skorelowane. Konieczne było więc podjęcie działań naprawiających tę sytuację, które omówimy dokładniej w rozdziale "Wybieranie zmiennych".

3 Inżynieria cech

3.1 Dodane zmienne

Po dokładnym wgłębieniu się w opisy poszczególnych plików odkryliśmy, że możemy bezboleśnie dodać do zbioru `X` jeszcze jedną kolumnę pochodzącą z plików `subject_train.txt` oraz `subject_test.txt`. Określa ona, który z 30 ochotników odpowiada za daną obserwację.

Dało nam to jednak kolumnę o trzydziestu wartościach. Nie mogliśmy jej potraktować jako numerycznej, ponieważ nie wiedzieliśmy nic na temat zależności pomiędzy uczestnikami. Z drugiej strony, większość modeli klastrujących nie akceptuje faktorów o więcej niż dwóch poziomach. Narzucającym się rozwiązaniem był one-hot-encoding, w rezultacie działania którego otrzymaliśmy 30 nowych kolumn w miejsce jednej.

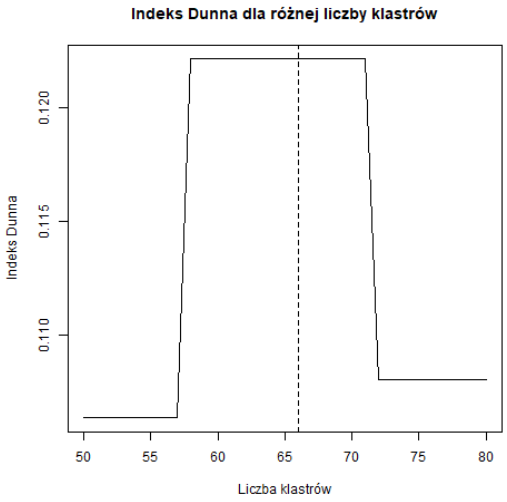
Tablica 2: Ilość obserwacji przypadających na uczestnika

Uczestnik	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Obserwacje	347	302	341	317	302	325	308	281	288	294	316	320	327	323	328
Uczestnik	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Obserwacje	366	368	364	360	354	408	321	372	381	409	392	376	382	344	383

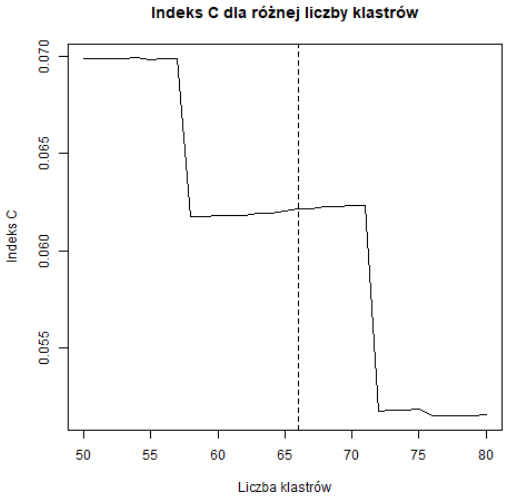
3.2 Wybieranie zmiennych

3.2.1 Correlation Clustering

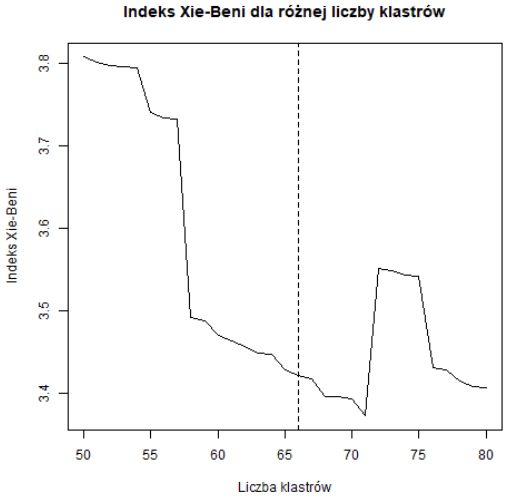
Z racji tego, że na wykresie korelacji zmiennnych były widoczne grupy skorelowanych zmiennych, postanowiliśmy zredukować wymiar danych stosując grupowanie po korelacji. W tym celu użyliśmy grupowania hierachicznego, w którym zamiast macierzy odległości podaliśmy zmodyfikowaną macierz korelacji (minus logarytm naturalny z wartości bezwzględnych). Następnie, by określić optymalną liczbę klastrów, użyliśmy 3 ideksów wewnętrzych określających jakość pogrupowania dla liczności grup od 50 do 81. Analizując wykresy 2, 3, 4 wybraliśmy 66 jako najlepszą liczbę grup.



Rysunek 2: Indeks Dunna(szukamy maksymalnych wartości)

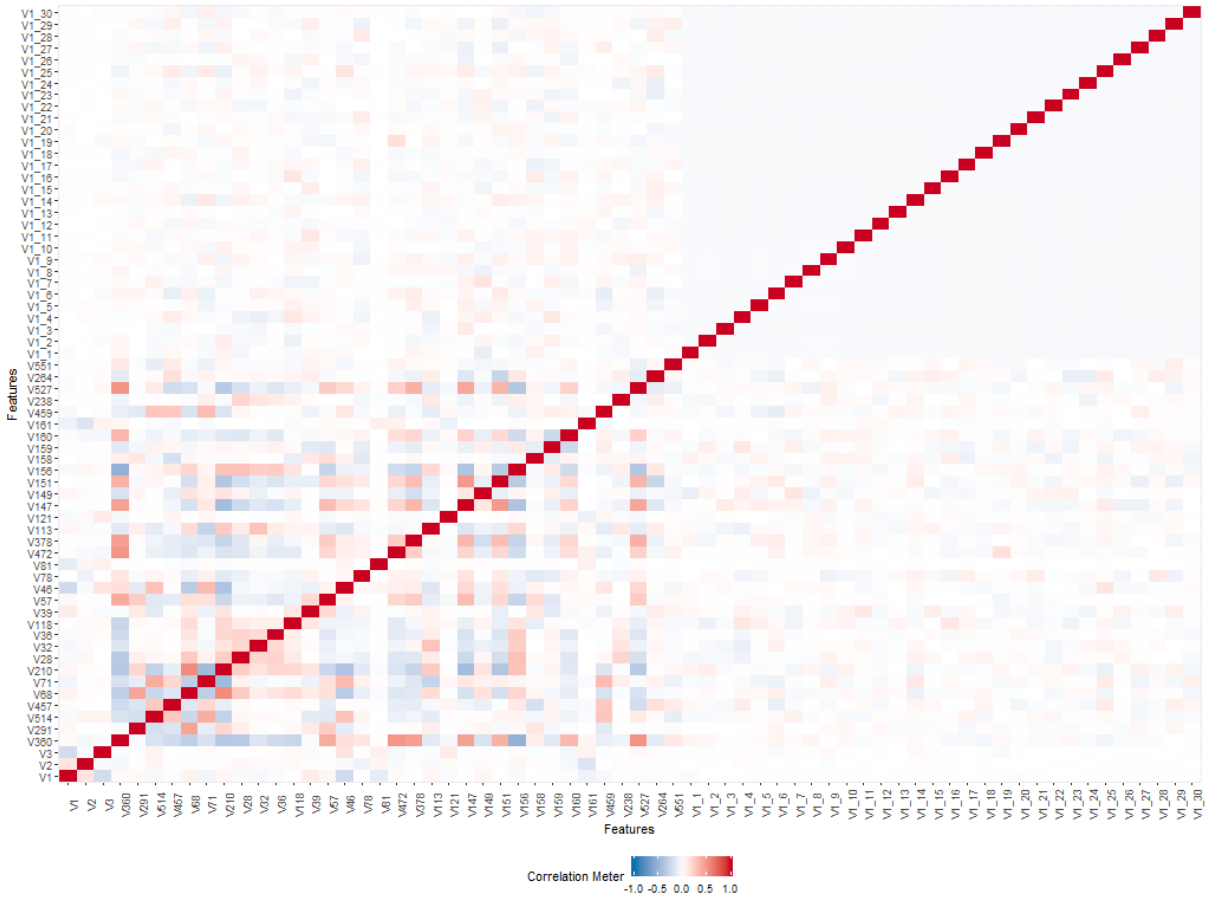


Rysunek 3: Indeks C(szukamy minimalnych wartości)



Rysunek 4: Indeks Xie-Beni(szukamy minimalnych wartości)

Mając już podział na grupy, wybraliśmy z każdej reprezentanta - zmienną, która miała największą średnią korelację z pozostałymi zmiennymi z danej grupy. Na wykresie 5 widać, że w nowym zbiorze danych nie ma już silnie skorelowanych grup zmiennych.



Rysunek 5: Macierz korelacji dla zbioru danych po correlation clustering

3.2.2 PCA

Jako inny sposób redukcji wymiarów sprawdziliśmy analizę głównych składowych(PCA). Użyliśmy do tego funkcji `dudi.pca()` z pakietu `ade4`. W obiekcie zwracanym przez tę funkcję znajdowała się znormalizowana macierz współrzędnych każdej obserwacji względem 50 najważniejszych składowych. Wybraliśmy tę macierz ze względu na bardzo dobry wynik dla klasteryzacji metodą k-średnich.

Tablica 3: Wyniki dla klasteringu metodą k-średnich

Miara	Dane z PCA	Surowe dane
Folkes-Mallows	0.70	0.56
Jaccard	0.52	0.39
Rand	0.86	0.84
precision	0.56	0.53
recall	0.88	0.58

4 Trenowanie modeli

Po wybraniu podzbiorów zmiennych za pomocą correlation clustering (podzbiór 66 zmiennych / kolumn, który będzie odtąd nazywany zbiorem CorrClust) i analizy głównych składowych (podzbiór 50 zmiennych / kolumn, który będzie odtąd nazywany zbiorem PCA) przyszła kolej na sprawdzenie, jak radzą sobie na nich algorytmy klasteryzujące, oraz na porównanie uzyskanych wyników z klasteryzacją na oryginalnym zbiorze danych zawierającym dodane przez nas opisane wcześniej zmienne. W ten sposób sprawdziliśmy czy odrzucenie niektórych zmiennych nie spowoduje zbyt znaczącej utraty informacji.

4.1 Testowane modele

Dla każdego ze zbiorów: oryginalnego, CorrClust i PCA zastosowaliśmy następujące algorytmy klasteryzujące z biblioteki mlr:

- Algorytmy bazujące na algorytmie k-means, pozwalające na narzucenie liczby klastrów:
  - cluster.kmeans
  - cluster.SimpleKMeans
  - cluster.XMeans
  - cluster.cmeans
- Algorytmy same dobierające optymalną liczbę klastrów:
  - cluster.FarthestFirst
  - cluster.Cobweb

oraz algorytmy z innych bibliotek:

- genie::hclust2 – algorytm bazujący na wartości indeksu Giniego, odporny na outliery
- cluster::pam – algorytm PAM (Partitioning Around Medoid) obierający obserwacje ze zbioru danych za centra klastrów

4.2 Sposób ewaluacji

Po wytrenowaniu ośmiu modeli klasteryzujących dla każdego z 3 zbiorów danych, trzeba było je porównać. Jako, że dysponujemy prawdziwymi etykietami tych zbiorów danych mogliśmy wykorzystać do tego celu indeksy zewnętrzne, mierzące jakość klasteryzacji. Dla każdej klasteryzacji obliczyliśmy następujące indeksy:

- Rand
- Jaccard
- Fowkles.Mallows
- Russel.Rao,

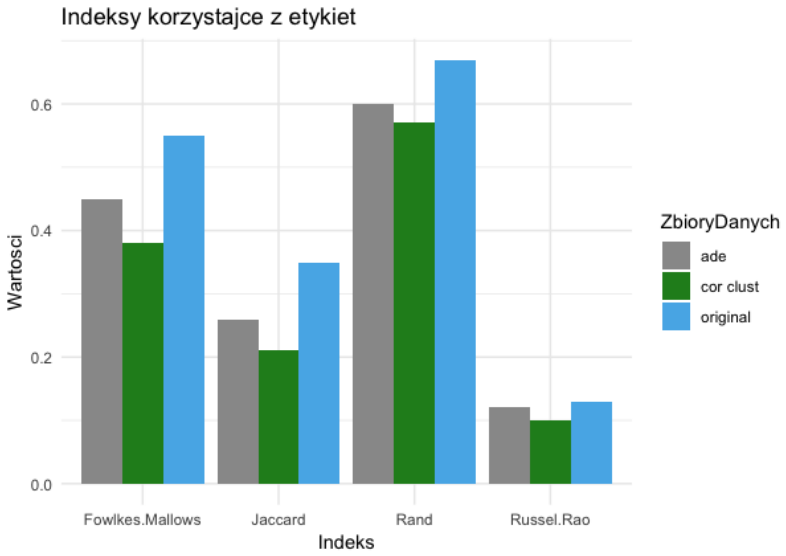
a także zapisaliśmy czas potrzebny dla danej klasteryzacji. W ten sposób zebraliśmy materiał do porównania różnych metod klasteryzacji dla każdego zbioru danych. Każdy z powyższych indeksów staramy się maksymalizować.

Przykładowa tabela, w której zbieraliśmy dane dla zredukowanego zbioru danych, wygląda następująco:

Tablica 4: Zebrane wartości indeksów i czasów dla oryginalnej ramki danych

Clustering Method	Time	Rand	Jaccard	Fowlkes.Mallows	Russel.Rao
cluster.kmeans	1.227	0.843	0.385	0.557	0.098
cluster.SimpleKMeans	5.332	0.664	0.333	0.577	0.168
cluster.XMeans	8.689	0.842	0.470	0.657	0.140
cluster.cmeans	4.238	0.764	0.285	0.453	0.094
cluster.FarthestFirst	4.350	0.172	0.169	0.410	0.168
cluster.Cobweb	229.588	0.169	0.169	0.411	0.169
genie::hclust2	19.206	0.857	0.463	0.639	0.123
cluster::pam	202.500	0.849	0.392	0.564	0.097

Następnie wyliczyliśmy średnią wartość każdego z indeksów dla każdego zbioru danych (po różnych metodach klasteryzacji) i porównaliśmy je na Rys.6.



Rysunek 6: Porównanie średniej wartości indeksów dla każdego zbioru danych

Na Rys.6 widać, że średnio metody klasteryzujące dają najlepsze wyniki na pełnym zbiorze danych. Drugie najwyższe wartości indeksów są otrzymywane na zbiorze PCA, natomiast zbiór danych otrzymany dzięki correlation clustering ma najniższe wartości indeksów dla różnych metod klasteryzacji. Uznaliśmy, że ograniczenie liczby zmiennych znacznie obniża jakość klasteryzacji i postanowiliśmy trenować końcowy model na całych danych, a nie ich podzbiorze.

5 Wybrany model

5.1 Podział na przewidziane 6 klastrów

Na tym etapie projektu porównaliśmy wyniki różnych modeli trenowanych na oryginalnym zbiorze danych, aby wybrać ten dający najlepsze rezulataty. Algorytmy cluster.FarthestFirst i cluster.Cobweb same decydują o optymalnej liczbie klastrów i zwracały klastrowania z jednym bądź dwoma klastrami. Dlatego też, mimo wysokich wartości porównywanych przez nas indeksów, postanowiliśmy je wykluczyć i skupić się na pozostałych algorytmach, dla których ustawiliśmy liczbę klastrów równą 6, jako że staramy się rozpoznać 6 różnych aktywności.

Algorytmy cluster.SimpleKMeans oraz (w mniejszym stopniu) cluster.cmeans miały niską wartość indeksu Randa, co zawęziło nasz wybór do agorytmów cluster.kmeans, cluster.XMeans, genie::hclust2 oraz cluster::pam. Z kolei niska wartość indeksu Jaccarda wykluczyła cluster.kmeans oraz cluster::pam i zawęziła nasze poszukiwania do dwóch kandydatów: cluster.XMeans oraz genie::hclust2. Analiza indeksów Fowlkesa-Mallowsa i Russela-Rao stoi jednogłośnie po stronie tego pierwszego. Nawet czas wykonania jest krótszy. Dlatego też optymalnym rozwiązaniem jest zastosowanie algorytmu cluster.XMeans z biblioteki mlr na całym zbiorze danych, zawierającym dodane przez nas zmienne.

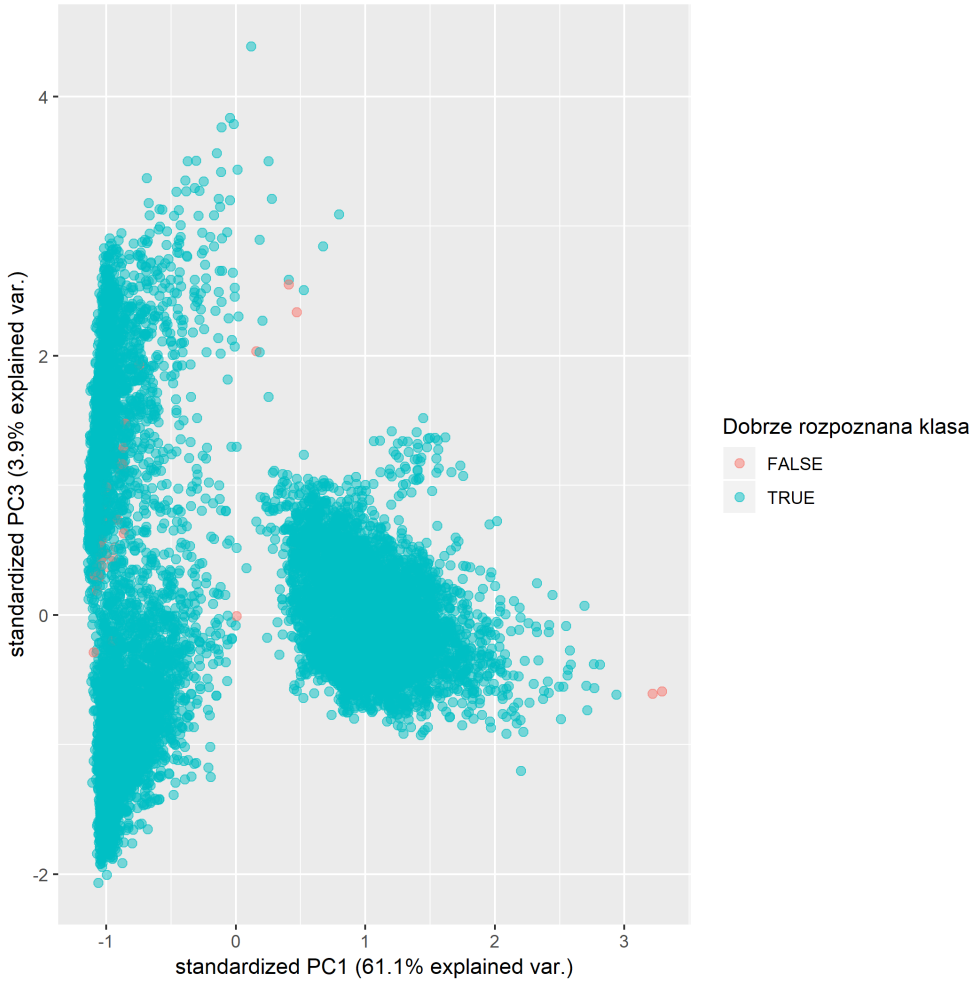
5.2 Podział na 3 klastry

Podczas analizy PCA widać było wyraźny podział na 2 lub 3 klastry(wykres 7). Pierwszy zawierał w sobie czynności oparte na ruchu(wchodzenie po schodach, schodzenie po schodach i po prostu chodzenie), a drugi czynności statyczne(leżenie, sidzenie, stanie). Drugi klaster dzielił się jeszcze na dwa pod względem pozycji ciała: pionowej(stanie, siedzenie) i poziomej(leżenie). Postanowiliśmy sprawdzić jak algorytm k-średnich podzieli zbiór uzyskany z PCA na 3 klastry(wykres 8)

Rysunek 7: 3 klastry w oryginalnych danych

Rysunek 8: 3 klastry w danych z PCA

Okazało się, że algorytm pomylił się tylko w 50 punktach na 10299 obserwacji(wykres 9). Można zatem stwierdzić, że metoda k-średnich na zbiorze danych z PCA bardzo dobrze dzieli aktywność fizyczną na 3 grupy: leżenie, stanie/siedzenie i ruch.



Rysunek 9: Zaznaczone punkty, które zostały przypisane do złego klastra

5.3 Podsumowanie

Rozpatrywany przez nas zbiór danych daje najlepsze rezutaty, gdy dzielimy go na trzy klastry odpowiadające aktywnościom dynamicznym (chodzenie, wchodzenie po schodach, schodzenie ze schodów), aktywnościom statycznym pionowym (stanie oraz siedzenie) oraz leżeniu. Dla takiego podziału algorytm k średnich daje nam wynik o dokładności (Eng. accuracy) 0,995.

Jeżeli jednak zależy nam na rozpoznaniu 6 aktywności i poszukujemy 6 klastrów im odpowiadających, wtedy najlepsze rezultaty osiągniemy dzięki algorytmowi cluster.XMeans z biblioteki mlr na całym zbiorze danych, zawierającym dodane przez nas zmienne.

Dodanie informacji o tym, kto wykonywał daną czynność znacznie polepszyło proces rozpoznawania aktywności, natomiast ograniczenie zmiennych za pomocą correlation clustering czy też za pomocą PCA nie pomogło w uzyskaniu lepszych rezultatów.

Literatura

[1] Dheeru Dua i Casey Graff, *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2017. http://archive.ics.uci.edu/ml, [dostęp: 15 czerwca 2019]