Human Activity Recognition with Smartphones

Raport podsumowujący projekt

Adrian Stańdo, Paweł Wojeciechowski, Kinga Ułasik

6 czerwca 2021 r.

Spis treści

1 Wprowadzenie

W ramach projektu z przedmiotu **Wstęp do uczenia maszynowego** stworzyliśmy model określający za pomocą klasteryzacji rodzaj ruchu wykonywanego przez 30 róznych osób na podstawie danych zebranych ze smartfona przymocowanego do pasa, a dokładniej z wbudowanego w telefon akcelerometru i żyroskopu.
Zostało uchwycone 3-osiowe przyspieszenie liniowe i 3-osiową prędkość kątową ze stałą częstotliwością 50 Hz. Każdy z czujników mierzy wartości (akcelerometr w jednostkach g - $9.81 m/s^2$, żyroskop w rad/s) 50 razy na sekundę dla każdej z trzech osi XYZ. Surowe dane pogrupowano w 2.56s okna (czyli 128 punktów pomiarowych), które nachodzą na siebie w 50% (czyli w połowie okna pomiarowego kolejne punkty pomiarowe będą należeć również dla kolejnego okna), uzyskujac w ten sposób 11500 obserwacji.
Rodzaje czynności to: schodzenie w dół, chodzenie do góry, schodzenie, siedzenie, stanie, leżenie.
Niniejszy raport dokumentuje naszą pracę związaną z tym zbiorem danych.

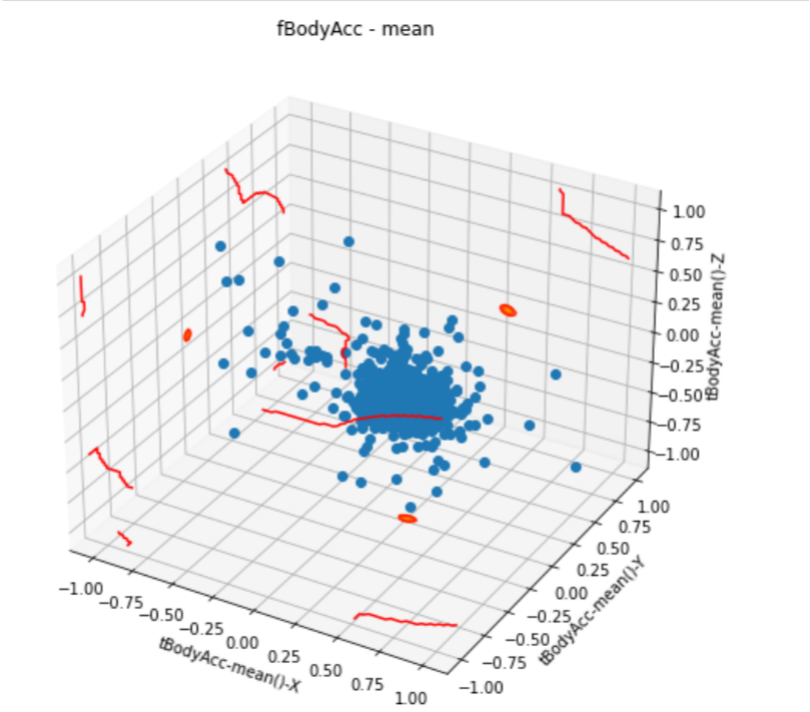
2 Eksploracja danych (EDA)

Pierwszy rzut oka na dane

Dane zostały przeskalowane od -1 do 1 oraz każdy z wyodrębnionych szeregów czasowych został przetworzony przez filtry odszumiające oraz oddzielające przyśpieszenie ruchu człowieka od grawitacyjnego, wyliczono również dodatkowe parametry, takie jak zryw czy normę wartośći oraz zastosowano szybką transformatę Fouriera.

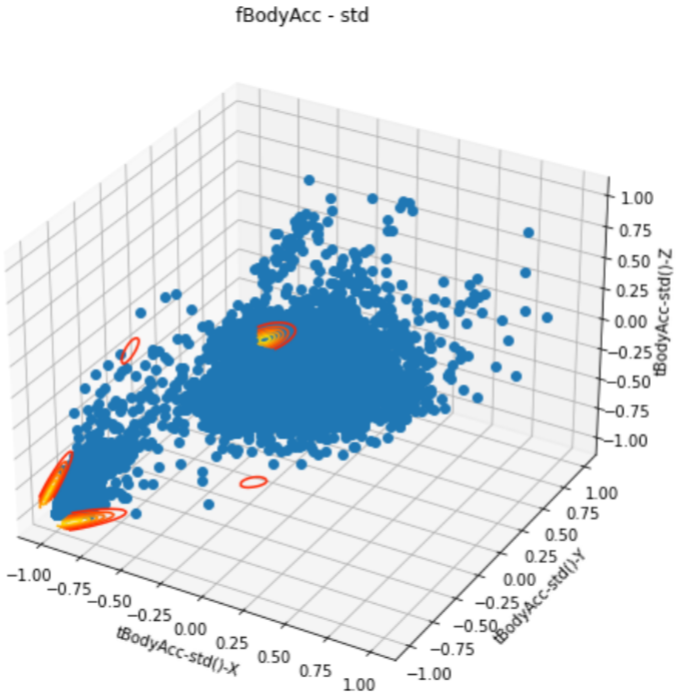
Następnie, spojrzeliśmy na trójwymiarowy rozkład dwóch podstawowych statystyk: średniej i odchylenia standardowego dla wielkości fBodyAcc przedwione na rys. 1 oraz rys. 2.

Rysunek 1: Rozkład średniej fBodyAcc



Większość obserwacji skupiona jest wokół środka układu współrzędnych. Taki widok świadczy o występowaniu outlierów, które są widoczne na wykresie jako pojedyncze punkty poza "środkiem".

Rysunek 2: Rozkład odchylenia standardowego fBodyAcc

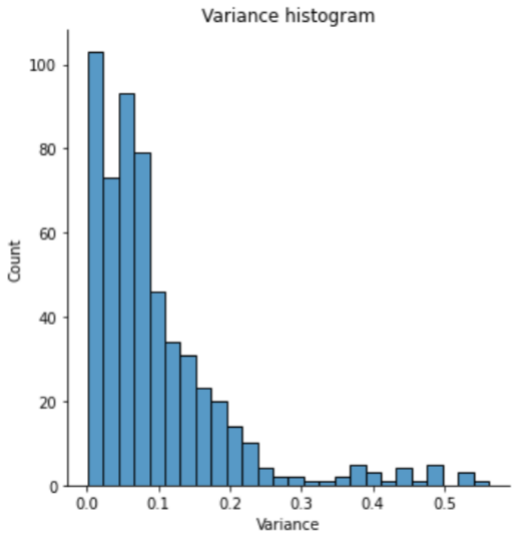


Na wykresie można wyróżnić dwie grupy. Jedna grupa oznacza aktywności cechujące się małą zmiennością przyspieszenia w każdym z kierunków - może to być np. siedzenie, a a druga grupa jest bardziej rozproszona, co może oznaczać inne aktywności wymagające zmienności ruchów.

Badanie wariancji

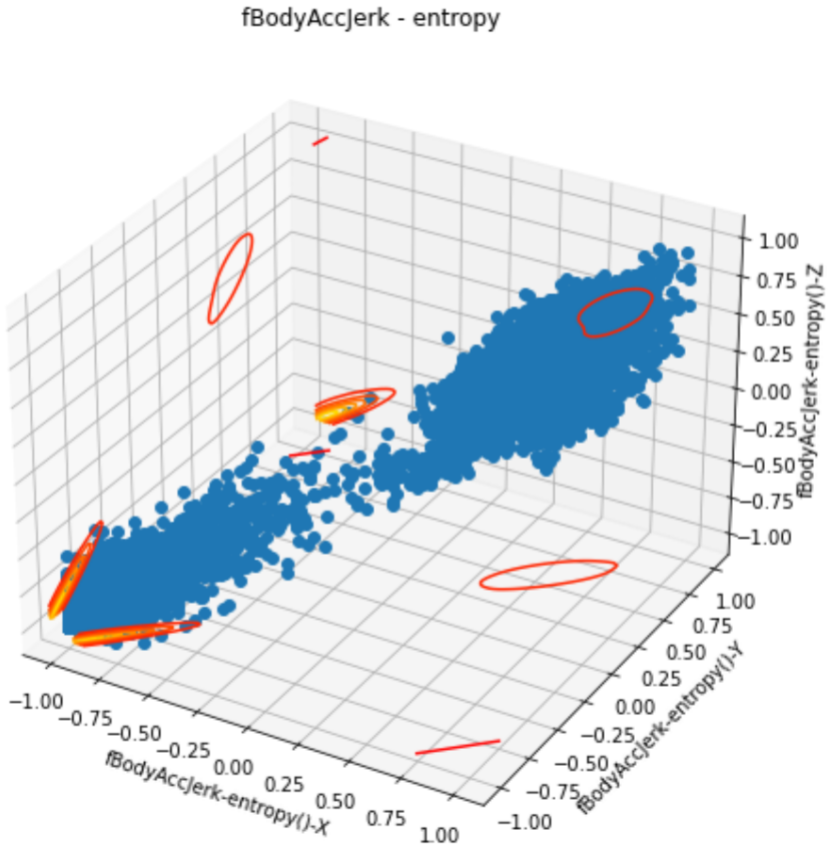
Następnie spojrzeliśmy na wariancję zmiennych, wizualizując jej wartości za pomocą histogramu widocznego na rys. 3.

Rysunek 3: Histogram wariancji zmiennych



Następnie, przyjrzelismy się kolumnom o największej wariancji. Zauważyliśmy, że fBodyAccJerk-entropy() w płaszczyznach X i Y mają największą wariancję w całym zbiorze danych, podczas gdy płaszczyzna Z jest na końcu powyższego zestawienia. Z tego powodu przyjrzeliśmy się rozkładowi tej zmiennej w trzech płaszczyznach.

Rysunek 4: Rozkład zmiennej fBodyAccJerk-entropy w trzech płaszczyznach



Na rys. 4 widać dokładnie dwie grupy, które z pewności rozróżniają zbiór danych. Zapewne jest to ponownie podział na aktywności śiedząceóraz te wymagające większych ruchów. Najwięcej obserwacji skupionych jest blisko punktu $(-1, -1, -1)$, natomiast druga grupa jest słabiej zarysowana na wykresach zrzutowanych gęstości.

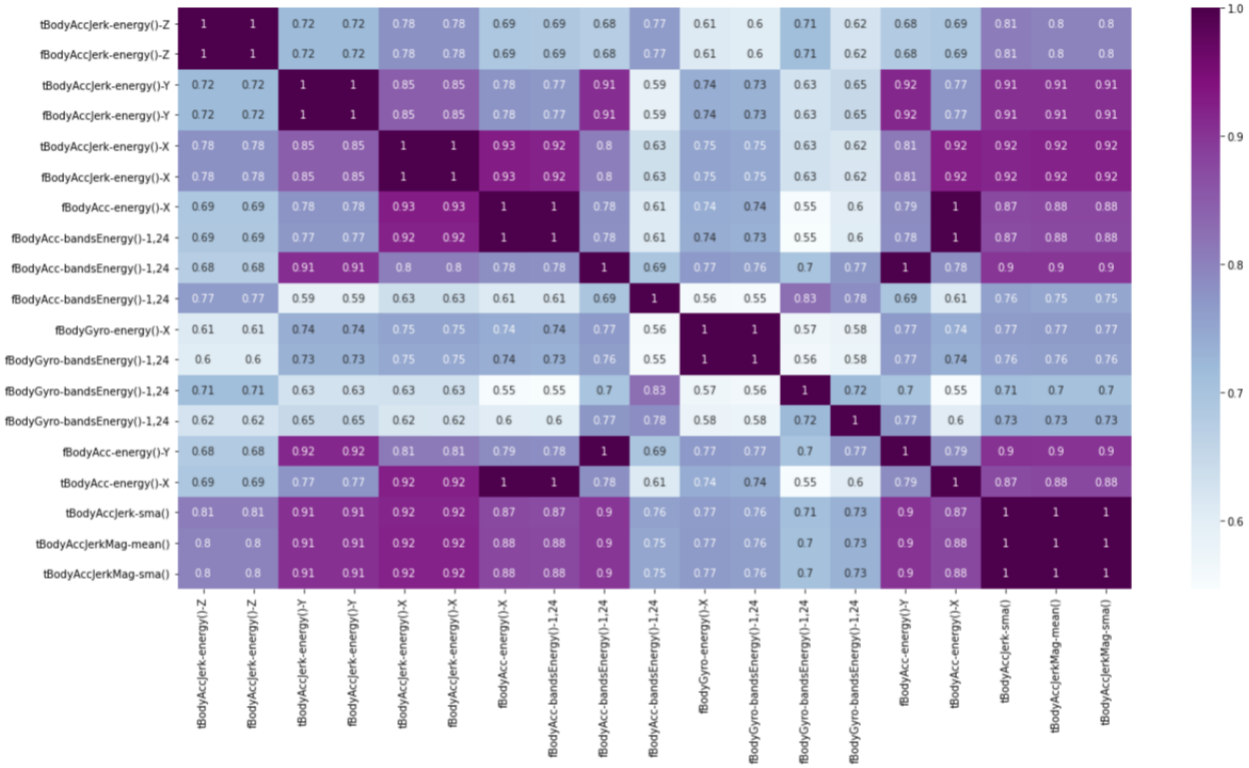
Badanie korelacji

Postanowiliśmy się przyjrzeć korelacji między zmiennymi, a dokładniej 10 największym korelacjom. Na początku, otrzymaliśmy same korelacje o wartości 1, co nas dosyć zastanowiło. Jednak, po przyjrzeniu się bliżej kolumnom okazało się, że tak wysoka korelacja wynika z normy euklidesowej nakładanej na niektóre zmienne.
Odfiltrowaliśmy więc wszystkie "jedynki", wynikową tablę widać na rys. 5, a heatmapa korelacji zmiennych z tabeli jest przedstawiona na rys. 6.

Rysunek 5: Tabela 10 największych korelacji po odfiltrowaniu "jedynek"

```
Top Absolute Correlations
tBodyAccJerk-energy()-Z  fBodyAccJerk-energy()-Z          1.000000
tBodyAccJerk-energy()-Y  fBodyAccJerk-energy()-Y          1.000000
tBodyAccJerk-energy()-X  fBodyAccJerk-energy()-X          0.999999
fBodyAcc-energy()-X      fBodyAcc-bandsEnergy()-1,24      0.999864
fBodyGyro-energy()-X     fBodyGyro-bandsEnergy()-1,24     0.999773
fBodyAcc-energy()-Y      fBodyAcc-bandsEnergy()-1,24      0.999635
tBodyAcc-energy()-X      fBodyAcc-energy()-X              0.999633
tBodyAccJerk-sma()       tBodyAccJerkMag-mean()           0.999615
                         tBodyAccJerkMag-sma()            0.999615
tBodyAcc-energy()-X      fBodyAcc-bandsEnergy()-1,24      0.999514
dtype: float64
```

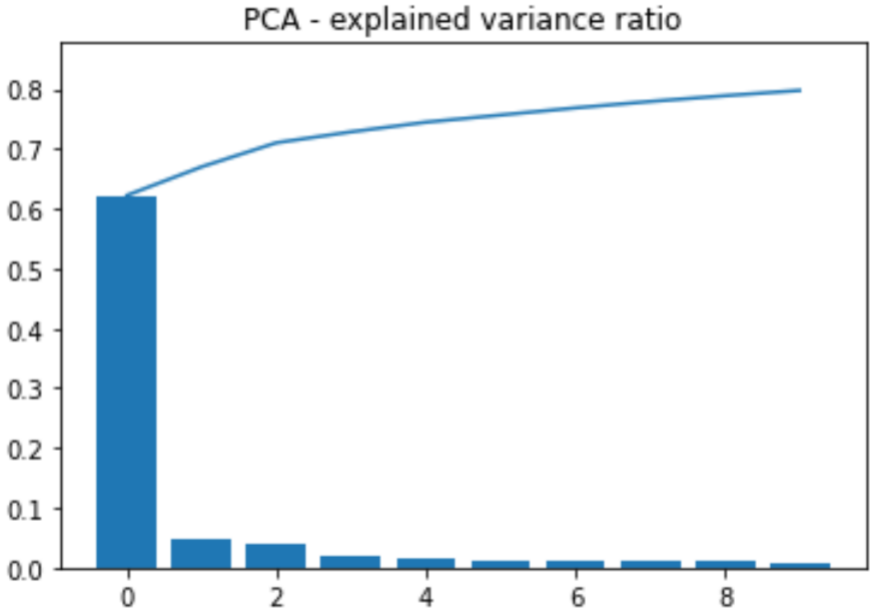
Rysunek 6: Heatmapa korelacji zmiennych o najwyższej korelacji



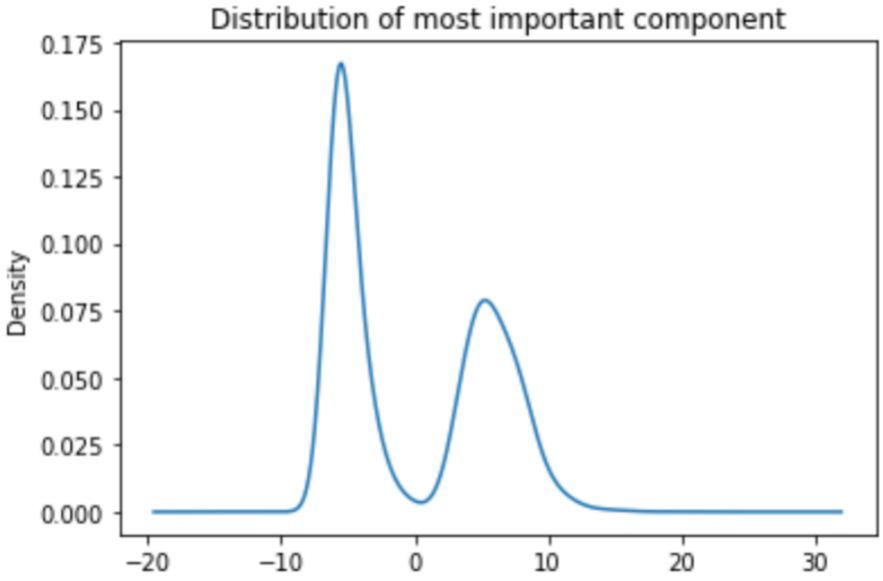
PCA

Na końcu eskploracji, za pomocą PCA znaleźliśmy kolumny, które najbardziej różnicują zbiór danych. Wynikowe wizualicje są przedstawione na rys. 7 oraz rys. 8.

Rysunek 7: Wytłumaczana wariancja w zależności od ilości komponentów PCA



Rysunek 8: Rozkład najważniejszego komponentu PCA

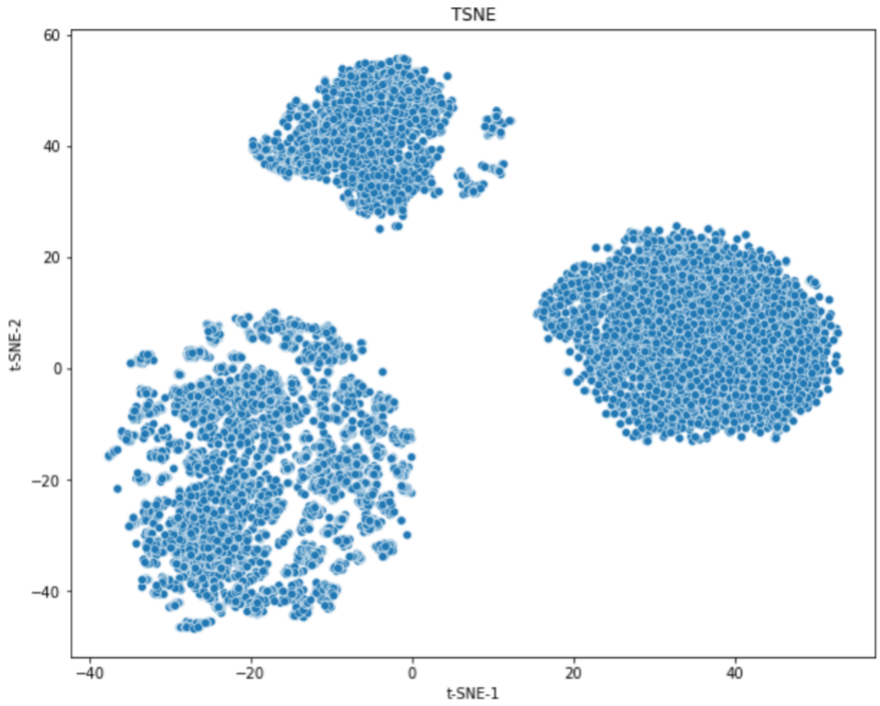


3 Redukcja wymiarów i pierwsze próby klasteryzacji

t-SNE

Pierwszym narzędziem jakie użyliśmy w celu redukcji wymiarów, było t-SNE. Udało nam się łatwo wyróżnić trzy klastry przedstawione na rys. 9. Dodatkowo, potem udało nam się jeszcze rozbić jeden klaster na dwa na pomocą analizy wartości niektórych zmiennych (głównie tych opisanych w sekcji powyżej).

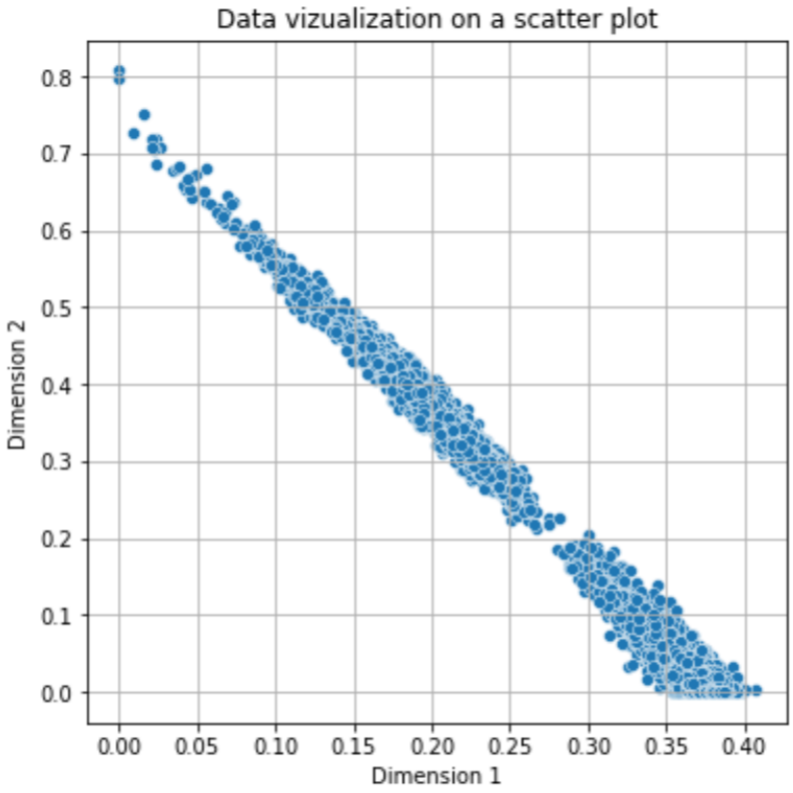
Rysunek 9: Klastry otrzymane za pomocą t-SNE



NMF

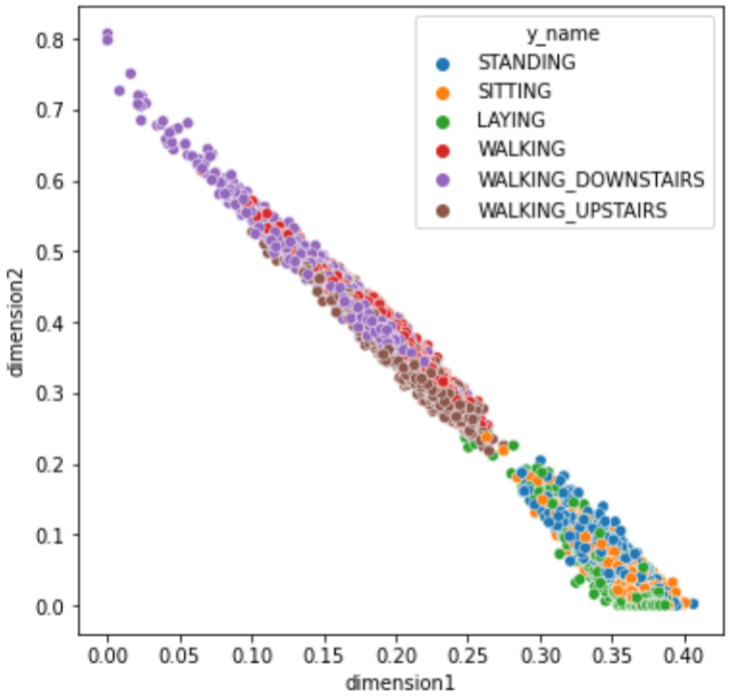
Kolejnym narzedziem, którego użyliśmy w celu zmiejszenia wymiarowości było NMF (Non-negative Matrix Factorization). Żeby użyć NMF dane muszą by nieujemne. Nasze dane już są przeskalowane od (-1,1) więc aby wyeliminować wartości ujemne, dodaliśmy 1 do każdego rekordu w tabeli.
Po przeprowadzeniu trasformacji (ilość komponentów to 2) przedstawiliśmy otrzymane wyniki za pomocą wykresu punktowego widocznego na rys. 10.

Rysunek 10: Klastry otrzymane za pomocą NMF



Już na pierwszy rzut oka widoczne są dwa klastry, które po sprawdzeniu można interpretować jako aktywności stacjonarne (czyli stanie, siedzenie, leżenie) oraz ruchowe (chodzenie, wchodzenie po schodach). Podstawy do takich wniosków zostały przedstawione na rys. 11.

Rysunek 11: Porównanie klastrów otrzymanych za pomocą NMF z rzeczywistymi nazwami grup obserwacji



KernelPCA

Ostatnim użytym przez nas narzędziem redukcji wymiarów było ponownie PCA, ale tym razem przyrzeliśmy się różnym jądrom.
Jako wyznaczniki działania redukcji wymiarowości, uznaliśmy wynik podstawowego algorytmu k-means. W ten sposób można zmierzyć jak dobrze sprawdza się PCA w swoim zadaniu, czyli ułatwieniu klasteryzacji.

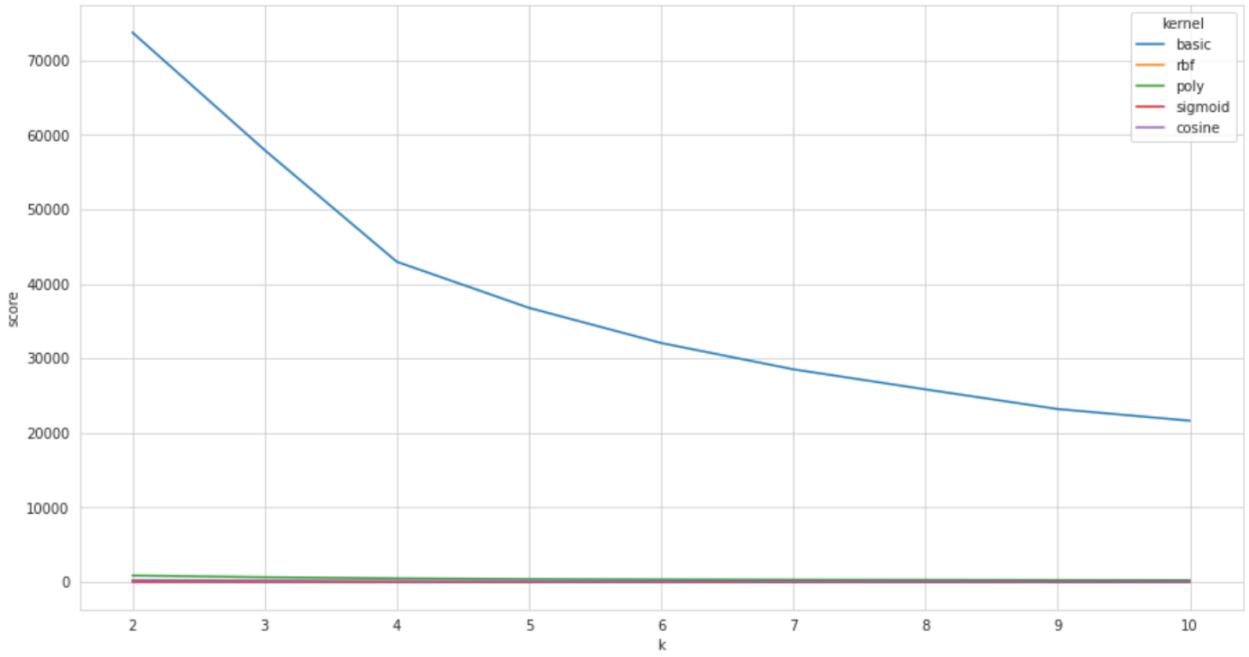
Sprawdziliśmy działania różnych jąder. Dla każdego jądra algorytm k-means został uruchomiony dla liczby klastrów $k = [2, 10]$ i zostały wyznaczone:

- średnia wartość współczynnika silhouette
- bezwładność

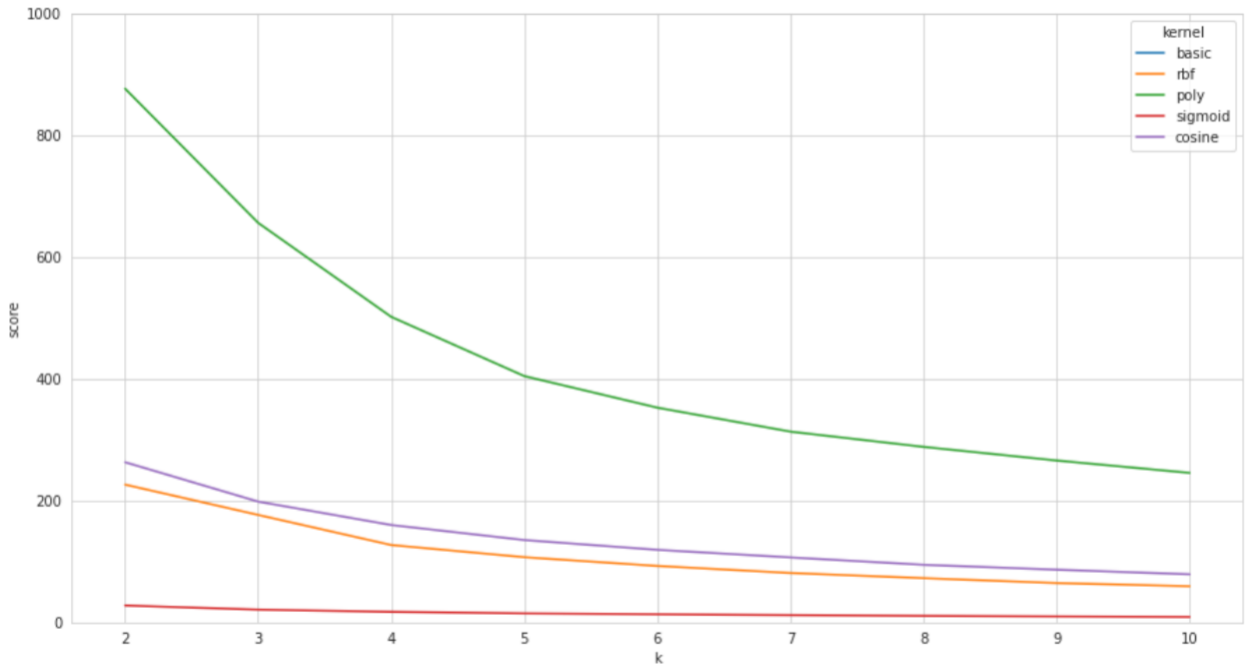
Bezwładność

Jak widać na rys. 12, algorytm k-means z podstawową wersją PCA osiąga o rząd wielkości większą bezwładność niż te, które posiadały PCA z nieliniowym jądrem. Po przybliżeniu wykresu (przybliżenie widoczne na rys. 13) można odczytać wartości dla nieliniowych jąder. Spośród nich najgorzej wypadło jądro wielomianowe. Jądra gaussowskie, cosine i sigmoid mają znacznie mniej wyraźny punkt ugięcią.

Rysunek 12: Wartość bezwładności dla róznych kerneli dla PCA



Rysunek 13: Wartość bezwładności dla róznych kerneli dla PCA, przybliżona, bez widocznego jądra 'basic'

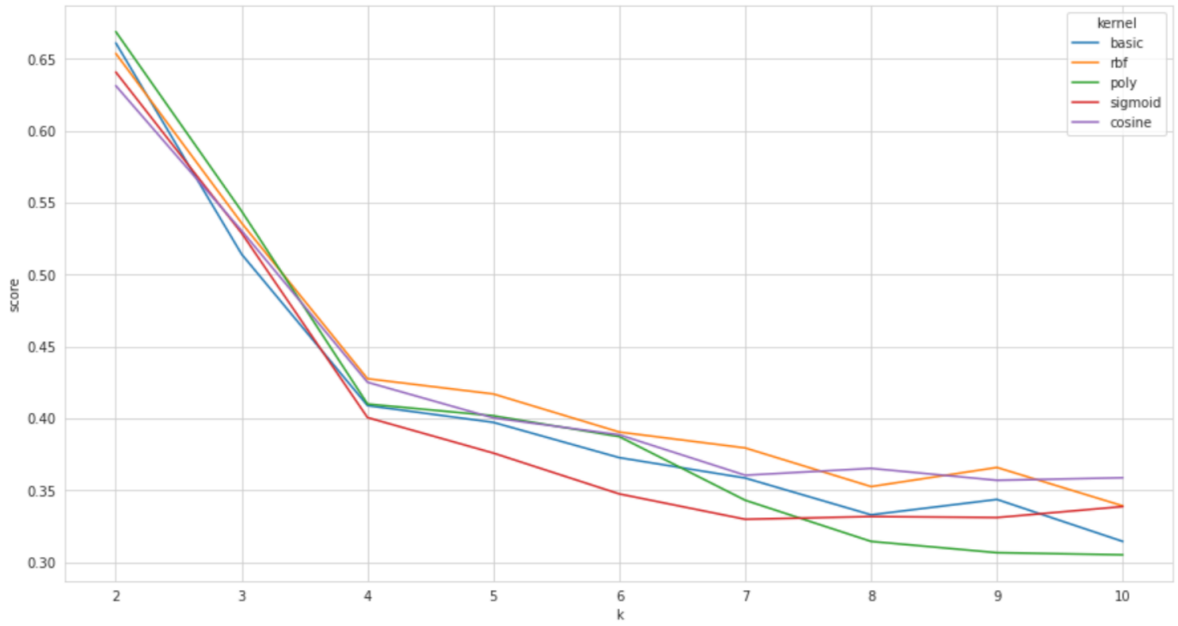


Współczynnik silhouette'a

W przypadku współczynnika silhouette'a wyniki są dużo bardziej zbliżone, widoczna jest tendencja gorszego wyniku przy większej liczbie klastrów.

Wszystkie algorytmy najlepsze wyniki dawały przy liczbie klastrów $k = 2$, co może wynikać z tego że obserwacje są głownie podzielone na dwa skupiska, czynności ruchowe i statyczne. Jeśli wybierać najlepsze jądro PCA na podstawie otrzymanych wyników, to faworytem wydaje się być jądro gaussowskie (rbf), które dla k>=3 osiąga najlepszy współczynnik sihouette'a oraz drugi najlepszy wynik w mierze bezwładności.
Wizualizaja będąca podstawą powyższych wniosków znajduje się na rys. 14

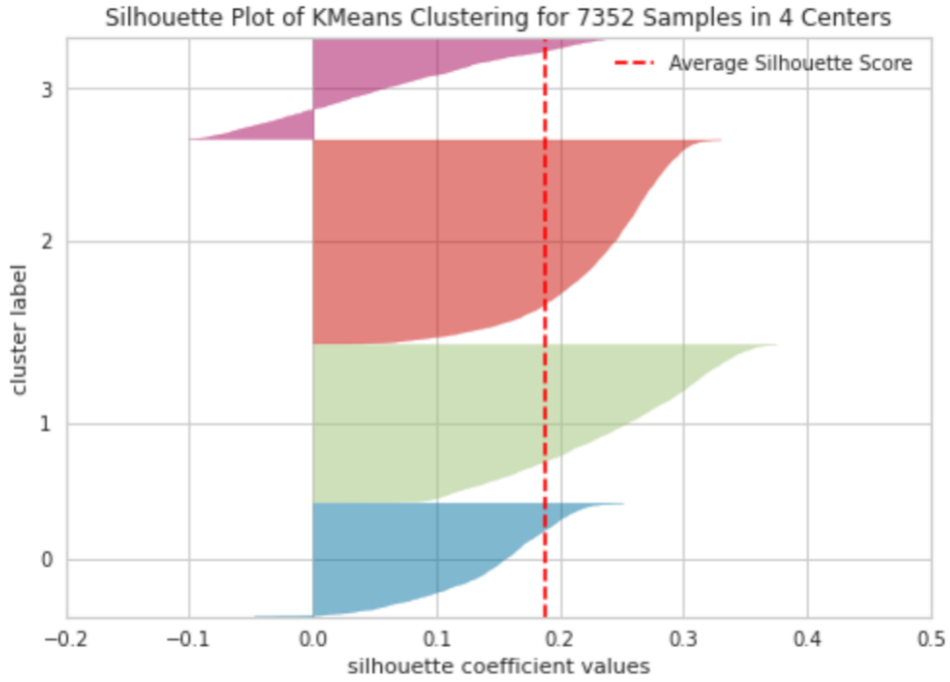
Rysunek 14: Wartość wpółczynnika silhouette'a dla róznych kerneli dla PCA



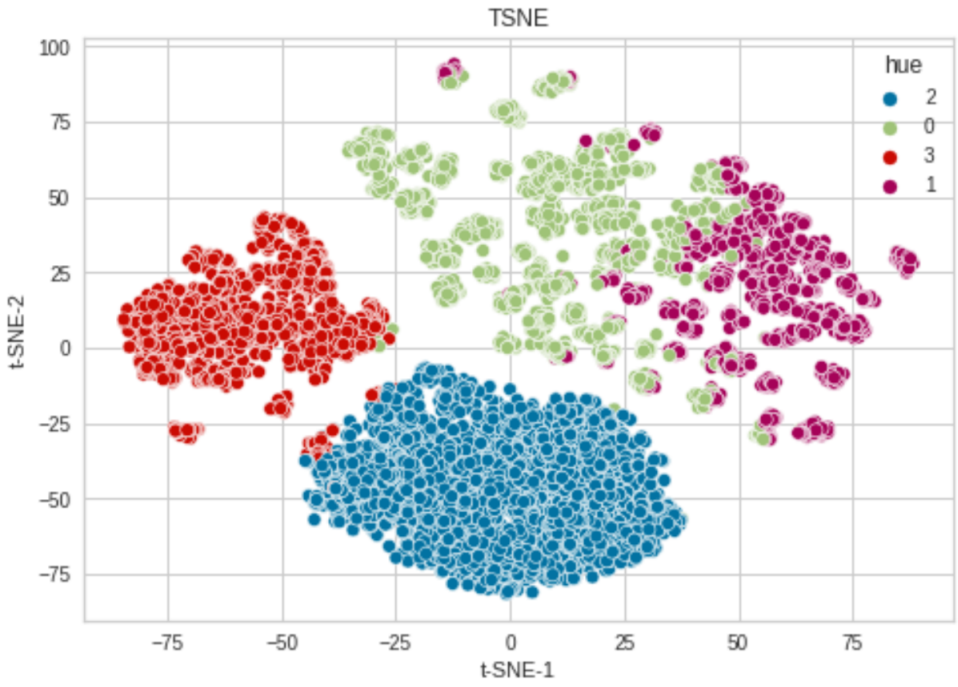
4 Finalny model

W celu wybrania finalnego modelu przetestowaliśmy kilka algorytmów, a dokładniej DBScan, KMeans, GMM oraz algorytm aglomeracyjny. Wybór padł na KMeans, ponieważ, naszym zdaniem najlepiej sobie poradził z klasteryzacją.
Po wybraniu algorytmów należało wybrać jeszcze ilość klastrów na jakies chcemy podzielić dane. Kierowaliśmy się wartością współczynnika silhouette'a i sensownością otrzymanych wyników zwizualizowanych za pomącą t-SNE. Algorytm najlepiej działał dla czterech klastrów. Trzy klastry to zbyt mała liczba, ponieważ w takim przypadku jeden klaster był bardzo liczny, podczas gdy w drugim nie ma żadnych obserwacji ze współczynnikim silhouette bliskim średniej (dla całego zbioru danych). Przy pięciu klastrach również wyniki nie były poprawne - w trzech z nich duża część obserwacji ma wartości ujemne współczynnika silhouette'a.
Analiza współczynnika dla czterech klastrów dla algorytmu KMeans jest przedstawiona na rys. 15, a końcowa wizualizacja za pomocą t-SNE na rys. 16.

Rysunek 15: Analiza wartości współczynnika silhouette'a dla ilości klastrów wynoszącej cztery dla algorytmu KMeans



Rysunek 16: Wizualizacja finalnej klasteryzacji



Na koniec, korzystając z oznaczeń klas rodzaju aktywności dla każdej obserwacji, za pomocą t-SNE porównaliśmy prawdziwy podział na grupy z otrzymaną przez nas klasteryzacją. Wizualizacja z prawdziwymi nazwami aktywności jest widoczna na rys. 15. Jak widać, niestety model nie jest w stanie odróżnić stania od siedzenia, za dobrze odseparowauje leżenie i czynności wymagające więcej ruchu. Niestety te ostanie nie do końca jest w stanie odpowiednio podzielić - chodzenie w ógre i w dół jest (mniej więcej) oddzielone od siebie, jednak samo chodzenie jest wmieszane w obie te grupy.

Rysunek 17: Wizualizacja obserwacji z realnymi nazwami aktywności