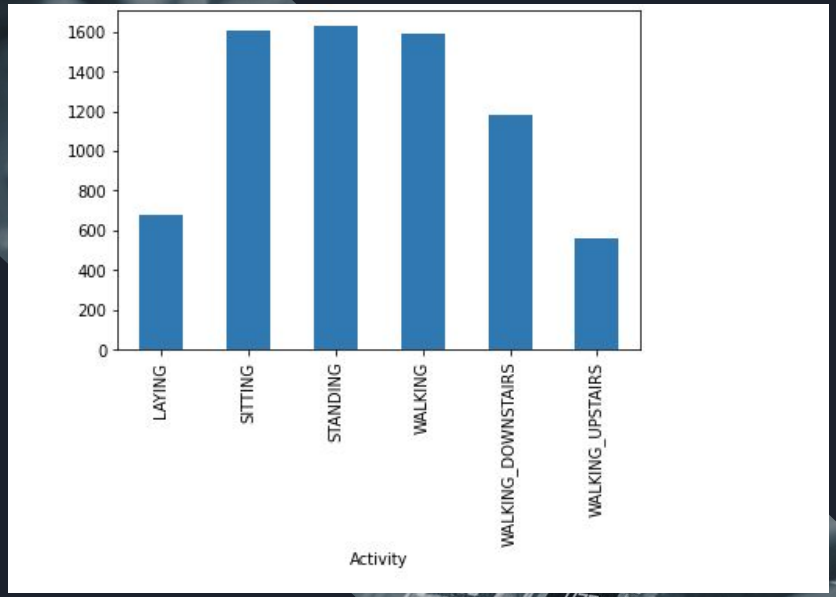
CoGOR-21+

B#L
AI

Jakub Ciborowski
Oliwia Gładysiak
Igor Kołakowski
Szymon Zborowski

Dane



Dane składają się z 6 klas aktywności

- **3** w spoczynku, **3** przemieszczające się
- Czynności wykonywane przez **30** osób

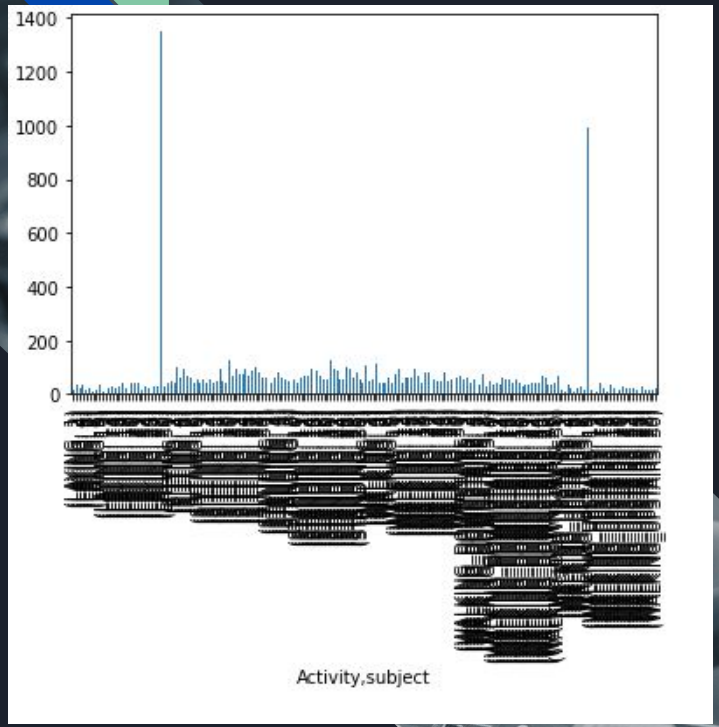
Dane zawierały wartości NaN

- **13%** wierszy zawierały NaN
- Istniały tylko dwie kolumny bez wartości NaN
- Ilość NaN w kolumnach była mniejsza niż **1%**

Podzieliliśmy na dane testowe i treningowe

- Dane treningowe nie były zbalansowane

Dane - przygotowanie przed trenowaniem



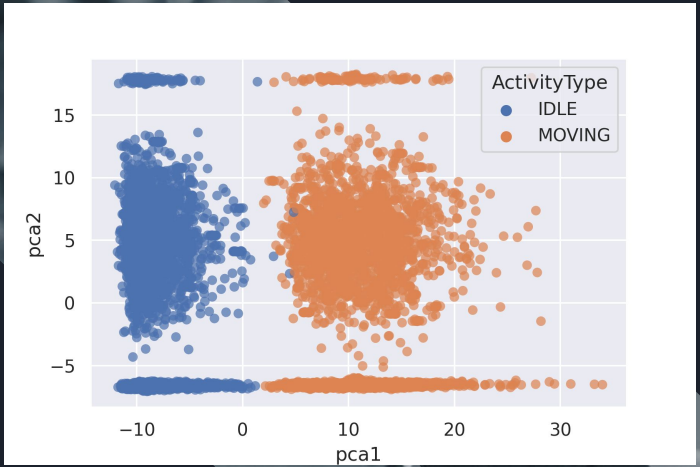
Kroki podjęte przed trenowaniem

- usunięcie kolumn ‘subject’, ‘void()’ oraz indexu
- usunięcie danych subjectów 28 i 9 - dane zaburzające

W początkowych eksperymentach uwzględniliśmy:

- redukcję wymiarowości
- wypełnianie braków danych

A co by było gdyby...?
czyli Regresja Liniowa



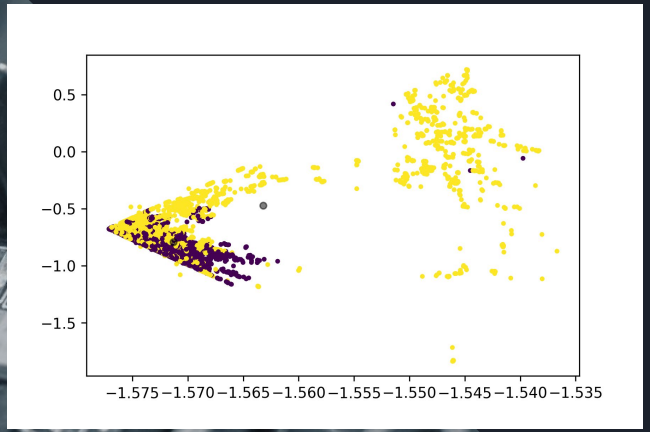
6 klas aktywności można w logiczny sposób podzielić na 2 klastry

Idle - { STANDING, SITTING, LAYING }

Moving - { WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS }

A co by było gdyby...?
czyli Regresja Liniowa

Dane	Precyzja dopasowania do 2 klastrów
Dane wyczyszczone z wyrównaną licznością danych	97.49 %
Dane wyczyszczone	96.79 %



Lasy losowe

	random forest with 10 trees with balanced weight	random forest with 20 trees with balanced weight	random forest with 40 trees with balanced weight	random forest with 80 trees with balanced weight	random forest with 100 trees with balanced weight	random forest with 150 trees with balanced weight	random forest with 200 trees with balanced weight	random forest with 500 trees with balanced weight
Accuracy	0.918699	0.897358	0.869919	0.887195	0.905488	0.906504	0.903455	0.90752
	random forest with 10 trees	random forest with 20 trees	random forest with 40 trees	random forest with 80 trees	random forest with 100 trees	random forest with 150 trees	random forest with 200 trees	random forest with 500 trees
Accuracy	0.886179	0.880081	0.878049	0.898374	0.888211	0.902439	0.906504	0.90752

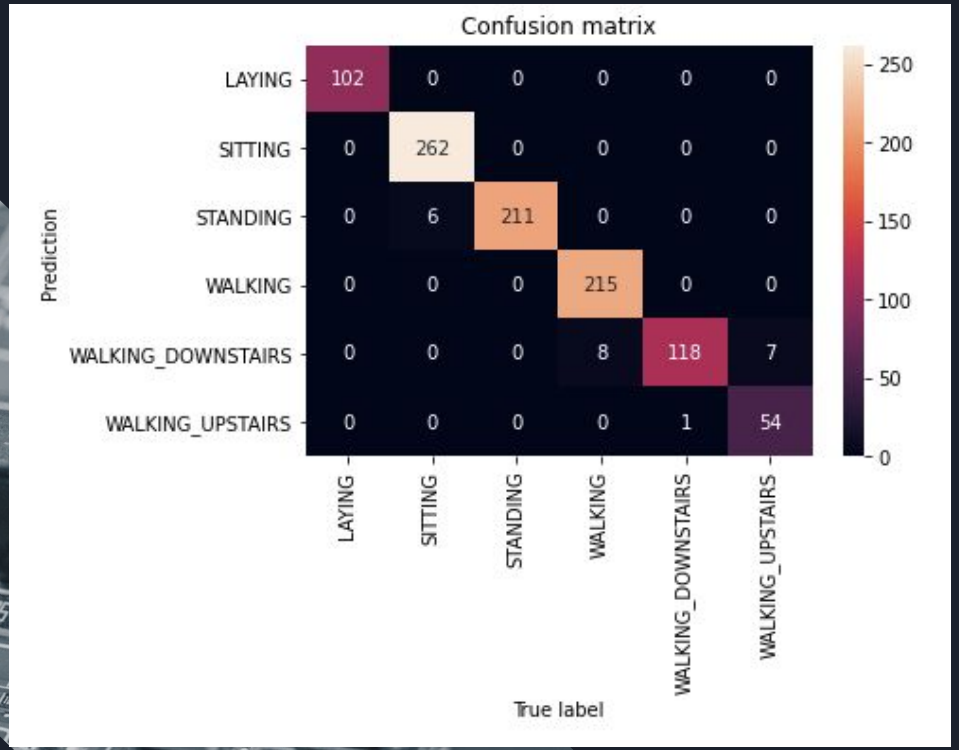
Lasy losowe

precision/ recall/ f1_score	random forest with 10 trees with balanced weight	random forest with 20 trees with balanced weight	random forest with 40 trees with balanced weight	random forest with 80 trees with balanced weight	random forest with 100 trees with balanced weight	random forest with 150 trees with balanced weight	random forest with 200 trees with balanced weight	random forest with 500 trees with balanced weight
LAYING	0.803/1.000/ 0.891	0.803/1.000/ 0.891	0.927/1.000/ 0.962	0.797/1.000/ 0.887	0.816/1.000/ 0.899	0.816/1.000/ 0.899	0.816/1.000/ 0.899	0.810/1.000/ 0.895
SITTING	0.931/0.851/ 0.889	0.979/0.851/0 .910	0.976/0.925/ 0.950	0.987/0.854/ 0.916	0.987/0.866/ 0.922	0.991/0.866/ 0.924	0.991/0.851/ 0.916	0.991/0.858/ 0.920
STANDING	0.928/0.919/ 0.924	0.932/0.976/0 .954	0.945/0.972/ 0.958	0.941/0.986 /0.963	0.941/0.986/ 0.963	0.941/0.991/ 0.965	0.925/0.991/ 0.957	0.937/0.991/ 0.963
WALKING	0.967/0.780/ 0.864	0.971/0.749/0 .846	0.933/0.749/ 0.831	0.974/0.830/ 0.896	0.979/0.830/ 0.898	0.984/0.825/ 0.898	0.979/0.821/ 0.893	0.979/0.830/ 0.898
WALKING_ DOWNSTAIRS	0.667/0.992/ 0.797	0.643/1.000/ 0.783	0.670/0.992/ 0.800	0.717/1.000/0 .835	0.708/1.000/0 .829	0.708/1.000/ 0.829	0.708/1.000/ 0.829	0.717/1.000/ 0.835
WALKING_ UPSTAIRS	0.978/0.738/ 0.841	1.000/0.754/ 0.860	0.979/0.770/0 .862	1.000/0.770/0 .870	1.000/0.754/0 .860	1.000/0.787/0 .881	1.000/0.787/0 .881	1.000/0.787/0 .881

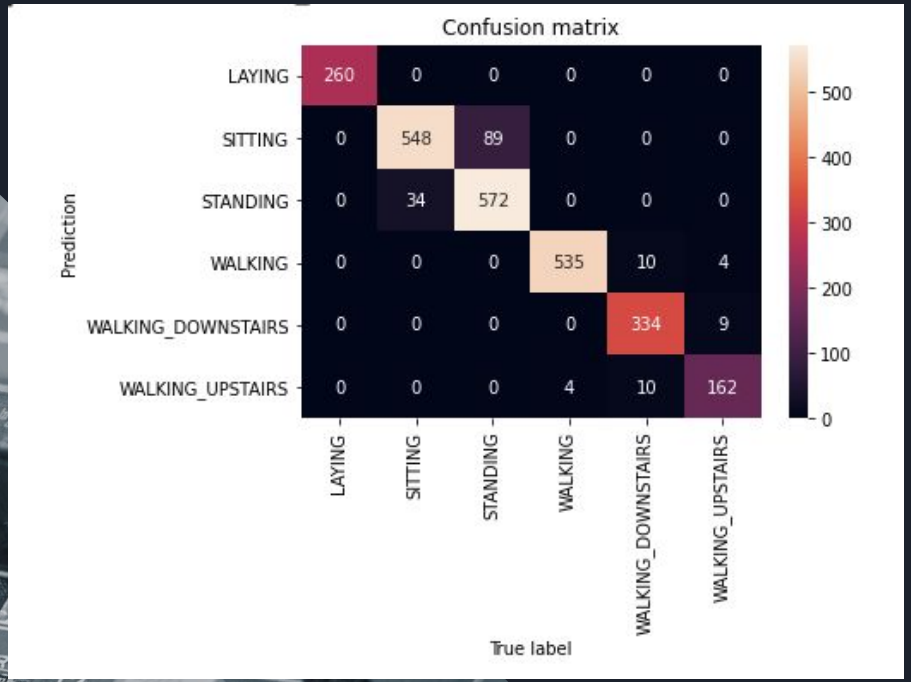
Lasy losowe

precision/ recall/ f1_score	random forest with 10 trees	random forest with 20 trees	random forest with 40 trees	random forest with 80 trees	random forest with 100 trees	random forest with 150 trees	random forest with 200 trees	random forest with 500 trees
LAYING	0.779/1.000 /0.876	0.791/1.000 /0.883	0.779/1.000 /0.876	0.779/1.000 /0.876	0.761/1.000 /0.864	0.761/1.000 /0.864	0.773/1.000 /0.872	0.767/1.000 /0.868
SITTING	0.953/0.840 /0.893	0.974/0.840 /0.902	1.000/0.843 /0.915	0.991/0.836 /0.907	0.991/0.836 /0.907	0.991/0.828 /0.902	0.996/0.847 /0.915	0.991/0.832 /0.905
STANDING	0.935/0.948 /0.941	0.928/0.972 /0.949	0.942/1.000 /0.970	0.933/0.991 /0.961	0.946/0.991 /0.968	0.937/0.991 /0.963	0.950/0.995 /0.972	0.937/0.991 /0.963
WALKING	0.948/0.812 /0.874	0.984/0.830 /0.900	0.978/0.807 /0.885	0.944/0.762 /0.844	0.989/0.821 /0.897	0.984/0.830 /0.900	0.989/0.830 /0.902	0.989/0.830 /0.902
WALKING_ DOWNSTAI RS	0.715/0.992 /0.831	0.720/0.992 /0.834	0.704/1.000 /0.826	0.654/1.000 /0.791	0.708/1.000 /0.829	0.713/1.000 /0.832	0.713/1.000 /0.832	0.713/1.000 /0.832
WALKING_ UPSTAIRS	0.894/0.689 /0.778	0.980/0.820 /0.893	0.980/0.803 /0.883	1.000/0.672 /0.804	1.000/0.820 /0.901	1.000/0.787 /0.881	1.000/0.803 /0.891	1.000/0.803 /0.891

XGBoost 3 uczestników zbiór testowy, 25 zbiór treningowy - 97.76%



XGBoost 8 uczestników zbiór testowy, 20 zbiór treningowy - 95.22%



XGBoost - cross validation bez podziału na osoby

```
[ ]  from sklearn.model_selection import StratifiedKFold
     from sklearn.model_selection import cross_val_score
     from sklearn.metrics import accuracy_score, classification_report
     import xgboost as xgb

     kfold = StratifiedKFold(n_splits=5, random_state=123, shuffle = True)
     model = xgb.XGBClassifier()
     results = cross_val_score(model, X, Y, cv=kfold)
```

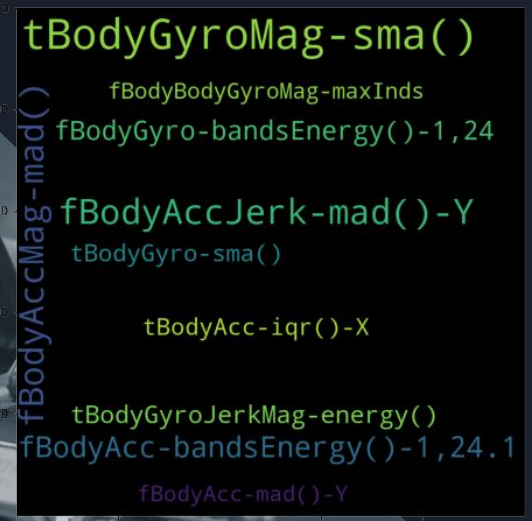
```
[ ]  results

     array([0.98727273, 0.9830303 , 0.99272286, 0.98787144, 0.98665858])
```

XGBoost

Dane	Precyzja
Dane czyste niezmodyfikowane	97.76 %
Dane czyste z wyrównaną licznością danych	97.35 %
Dane ze zmniejszonym rozmiarem zbioru	95.28 %

XGBoost - Importances

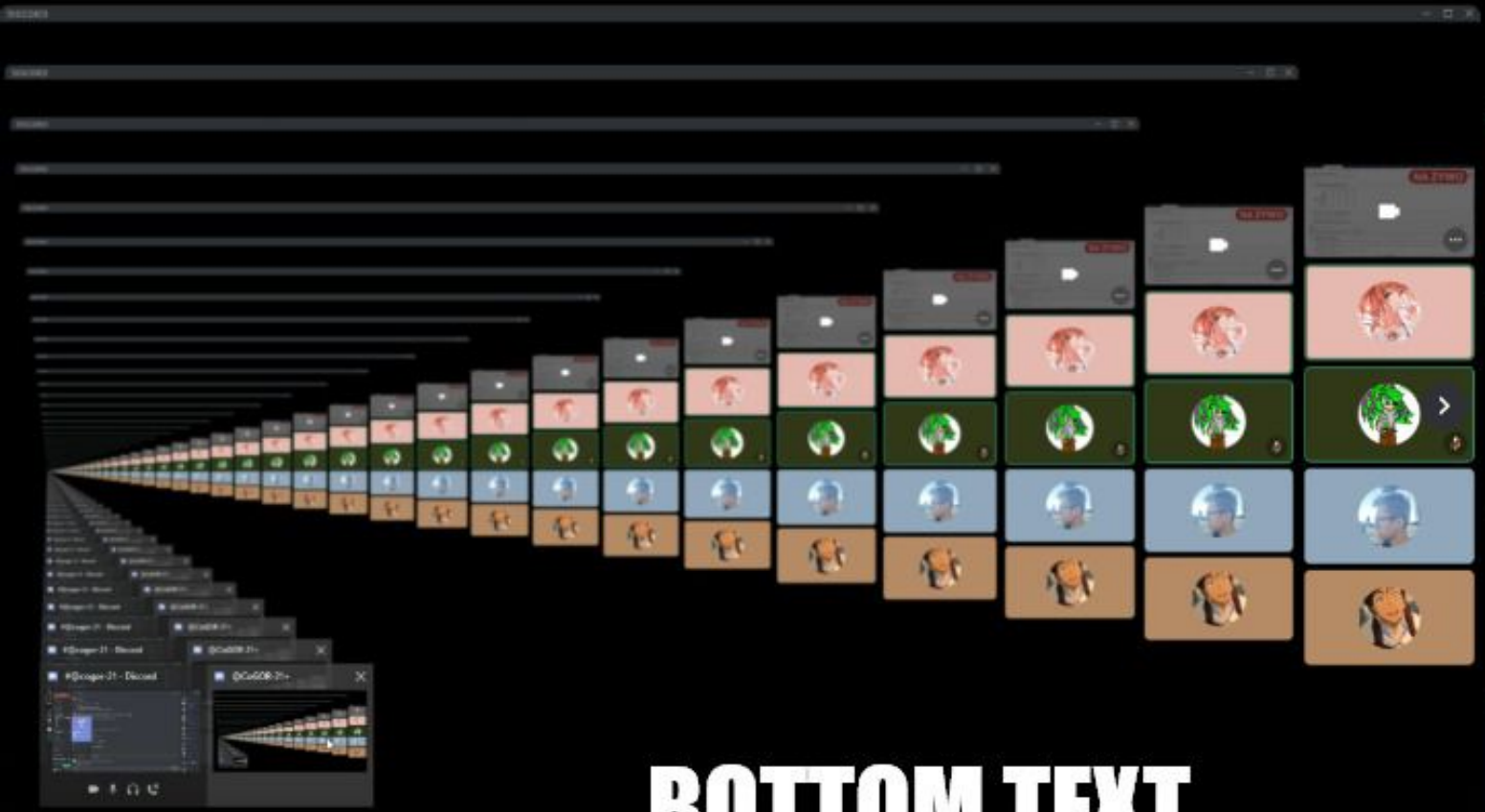
```
Sensor group: Body Acceleration (frequency)
Min in group: 0.008812, max in group: 410.929779, sum of values in group: 3950.171397, mean of values in group: 22.572408, Elements in group: 175
Sensor group: Body Acceleration
Min in group: 0.021953, max in group: 636.378494, sum of values in group: 3025.131423, mean of values in group: 21.154765, Elements in group: 143
Sensor group: Gravity Acceleration
Min in group: 0.050671, max in group: 162.244234, sum of values in group: 1321.470202, mean of values in group: 31.463576, Elements in group: 42
Sensor group: energy-mean
Min in group: 0.054298, max in group: 0.054298, sum of values in group: 0.054298, mean of values in group: 0.054298, Elements in group: 1
Sensor group: angle()
Min in group: 3.322535, max in group: 88.479849, sum of values in group: 112.693465, mean of values in group: 22.538693, Elements in group: 5
```



Podsumowanie

- Przedstawiliśmy modele o dużej **skuteczności**
- Zależało nam na możliwie **prostych** modelach
- Opcja skutecznej i bardzo taniej metody określenia stanu osoby jako **aktywna / w spoczynku**
- Charakter danych sprawia, że potencjalne błędy **nie są krytyczne**
- Dane jasno opisane, więc można pozyskiwać nowe i **rozwijać** modele

IGOR



BOTTOM TEXT