Human Gait Analysis

Human Activity Recognition from Wearable Inertial Sensor Networks

Mohamed NIANG

Staff ML Scientist

Plan

1. What & how you are solving it ...
2. State of the art
3. The development environment
4. DataSet Description / Exploration
5. Data Preprocessing & Feature Engineering
6. Models Tuning & Performance
7. Conclusion: Achieved results
8. Conclusion: Recap of the work
9. Next Steps / Roadmap

1. What & how you are solving it ...

Context of the research and problematic:

- Increase in the number of people with mobility impairments
- Availability of inertial unit data
- Need for connected medical devices to monitor the activities of the elderly (walking, falls, etc)
- Regularly monitor the human's activity (apple watch, etc.)
- model not robust to environmental changes
- Use of multiple sensors to solve the problem (not always efficient)

Goals:

- Use only one sensor
- Use a novel feature engineering approach
- use machine learning models

2. State of the art

Your AI tasks

- The inputs are data from the inertial unit with 3-axis accelerometers and 3-axis gyrometers
- The task is to predict the human activity (12) from these data using ML
- Benchmarks of some works on this field

Work	Data Generation	Evaluation Metrics	Validation Protocol
Pan et al.	Semi-Non-Overlapping-Window	Accuracy	Cross validation and Leave-One-Subjet-Out
Kim and Choi	Semi-Non-Overlapping-Window	Accuracy, F-measure	Unknown
Catal et al.	Unknown	Accuracy, AUC, F-Measure	10-fold cross validation

2. State of the art

Existing Solutions & how to use them for your task:

- Legacy methods & state of the art models: k-nearest neighbor, svm, random forest, cnn, rnn
- The results obtained from these studies range from an average of 98% accuracy
- Models used in this work:
    - logistic regression with lasso penalization (hyperparameter optimization included)
    - decision tree classifier with gridsearchCV
    - support vector classifier with gridsearchCV
    - k-nearest neighbors with gridsearchCV
    - random forest with gridsearchCV
    - xgboost with gridsearchCV

3. The development environment

The development environment

- Dev tools / IDE: Google Colab Pro
- Dev language: Python
- Python libraries: pandas, numpy, matplotlib, seaborn, scipy, sklearn, xgboost
- Google Colab Pro has more resources than the free version

The selected models

- The selected models is available through selected AI framework
- Model parameters are found via a gridsearchCV

4. DataSet Description / Exploration

DataSet Description:

ID	Activity	Time sec (min)	Percent	Samples	Description
1	Walking	11544 (192)	32.15	679073	Walking and turning at various speeds on a flat surface
2	Running	1218 (20)	3.39	71653	Running at various paces
3	Going up	2237 (37)	6.23	131604	Taking stairs up at various speeds
4	Going down	1982 (33)	5.52	116637	Taking the stairs down at various speeds and steps
5	Sitting	4111 (68)	11.45	241849	Sitting on a chair; sitting on the floor not included
6	Sitting down	409 (6)	1.14	24112	Sitting on a chair; sitting down on the floor not included
7	Standing up	380 (6)	1.06	22373	Standing up from a chair
8	Standing	5587 (93)	15.56	328655	Static standing on a solid surface
9	Bicycling	2661 (44)	7.41	156560	Typical bicycling
10	Up by elevator	1515 (25)	4.22	89144	Standing in an elevator while moving up
11	Down by elevator	1185 (19)	3.30	69729	Standing in an elevator while moving down
12	Sitting in car	3069 (51)	8.55	180573	Sitting while an travelling by car as a passenger
	Total	35903	598	100.00	2111962

Source: [GitHub]

4. DataSet Description / Exploration

DataSet Exploration:

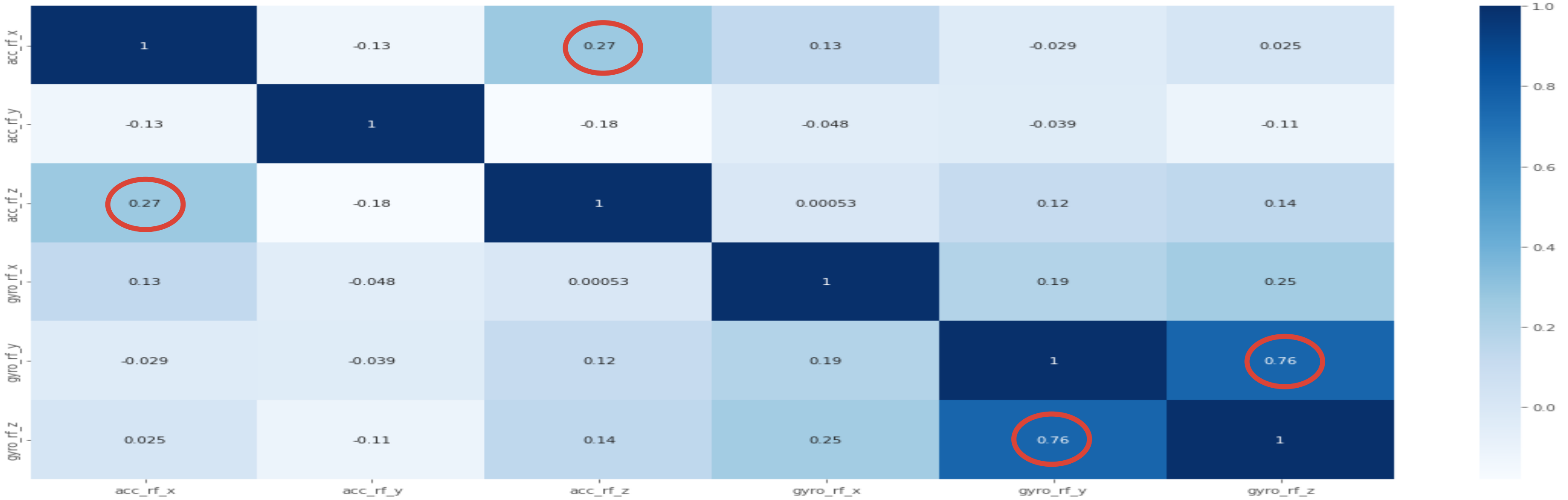
	acc_rf_x	acc_rf_y	acc_rf_z	gyro_rf_x	gyro_rf_y	gyro_rf_z	act	filename
0	-8232	-376.0000	13232.0000	-29.0000	181.0000	-167.0000	9.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
1	-8244	-236.0000	13252.0000	-146.0000	157.0000	24.0000	9.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
2	-8208	-88.0000	13188.0000	-207.0000	114.0000	33.0000	9.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
3	-8144	24.0000	13204.0000	-199.0000	167.0000	-23.0000	9.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
4	-8204	28.0000	13220.0000	-258.0000	193.0000	-124.0000	9.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
...	...	...	...	...	...	...	...	...
2111397	-9140	-3028.0000	13632.0000	105.0000	-61.0000	-155.0000	5.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
2111398	-9112	-2996.0000	13644.0000	93.0000	3.0000	-138.0000	5.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
2111399	-9124	-3036.0000	13664.0000	151.0000	-1.0000	-165.0000	5.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
2111400	-9124	-2984.0000	13616.0000	145.0000	-94.0000	-168.0000	5.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...
2111401	-9076	-2980.0000	13692.0000	91.0000	-95.0000	-105.0000	5.0000	/content/drive/MyDrive/myproject/data/HuGaDB_v...

2111402 rows × 8 columns

5. Data Preprocessing & Feature Engineering

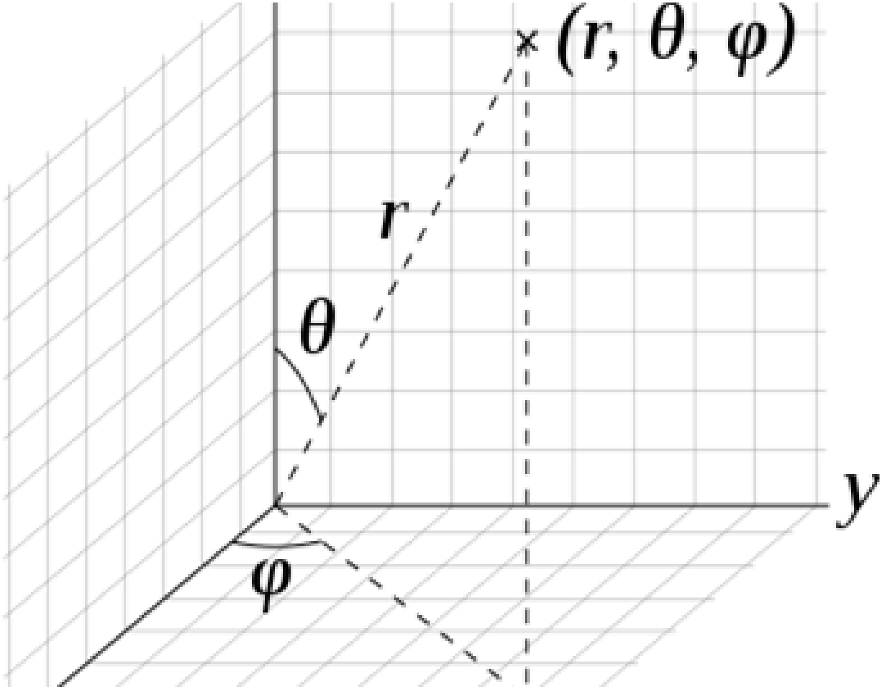
Data preprocessing

- Preprocessing: type correction, delete nan (5%) and inf values
- Correlation analysis:



5. Data Preprocessing & Feature Engineering

Features engineering



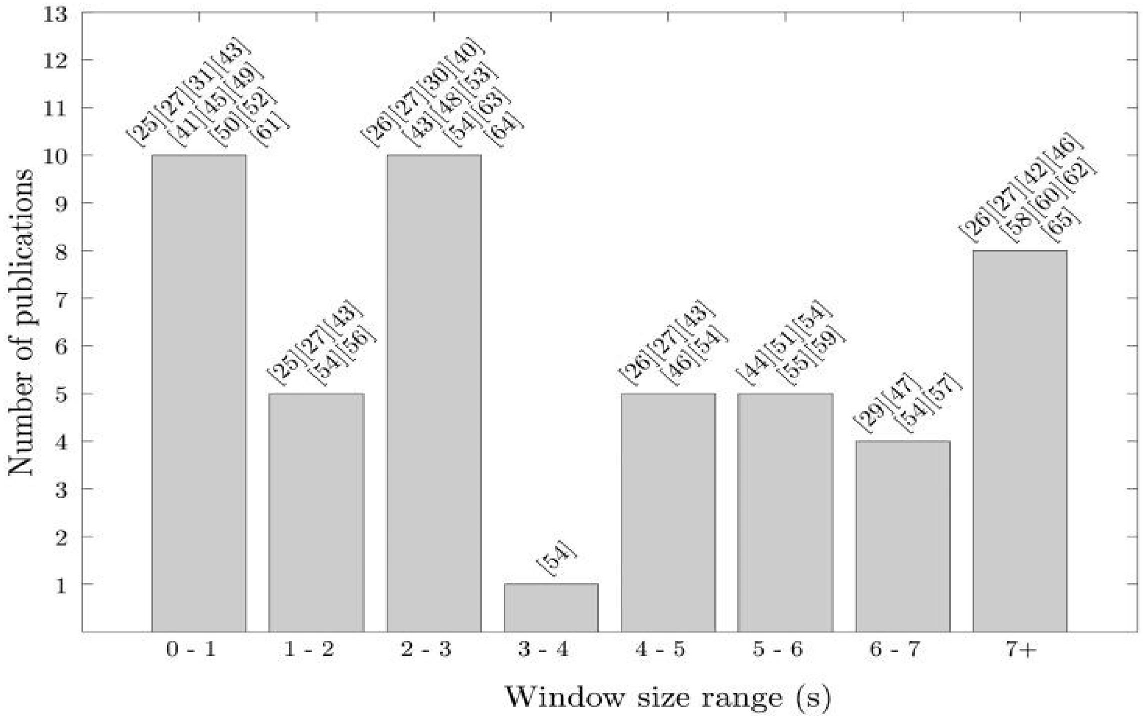
$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \arccos \frac{z}{\sqrt{x^2 + y^2 + z^2}} = \arccos \frac{z}{r} = \arctan \frac{\sqrt{x^2 + y^2}}{z}$$

$$\varphi = \begin{cases} \arctan(y/x) & \text{if } x \geq 0 \\ \arctan(y/x) + \pi & \text{if } x < 0 \end{cases}$$

5. Data Preprocessing & Feature Engineering

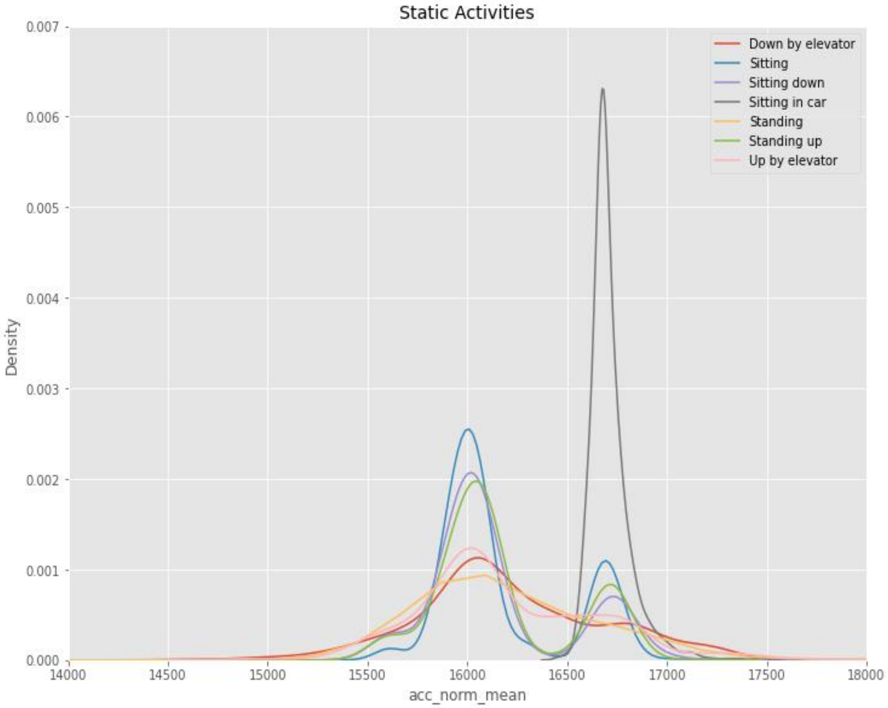
Data aggregation

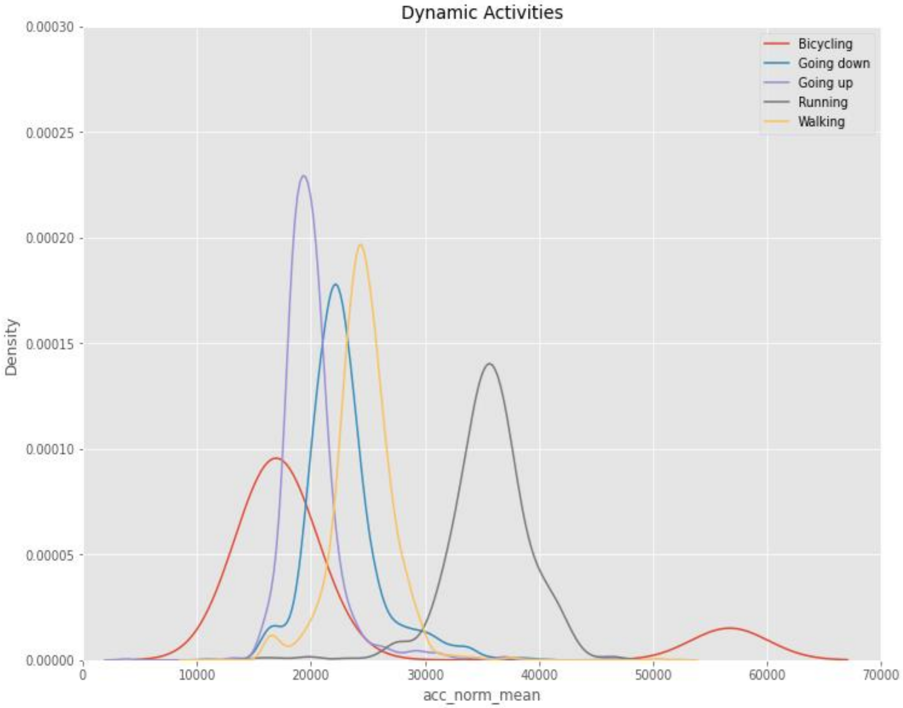


- Time window used: 3s
- Computed statistics: min, max, mean, median, std, skew, kurtosis, iqr, median absolute deviation, mean absolute deviation, range (min - max)
- Others features extraction methods: logabs and markov features (lagged features t-1 and t-2)

5. Data Preprocessing & Feature Engineering

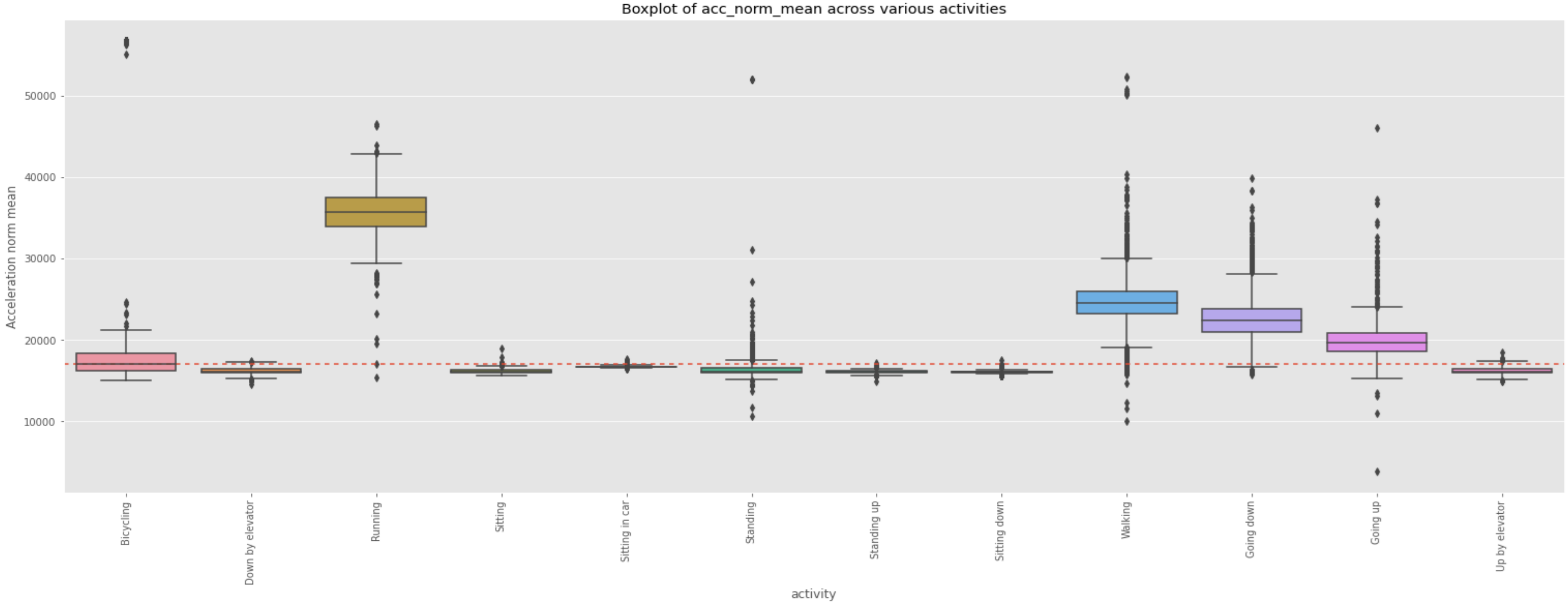
Features engineering results for distribution





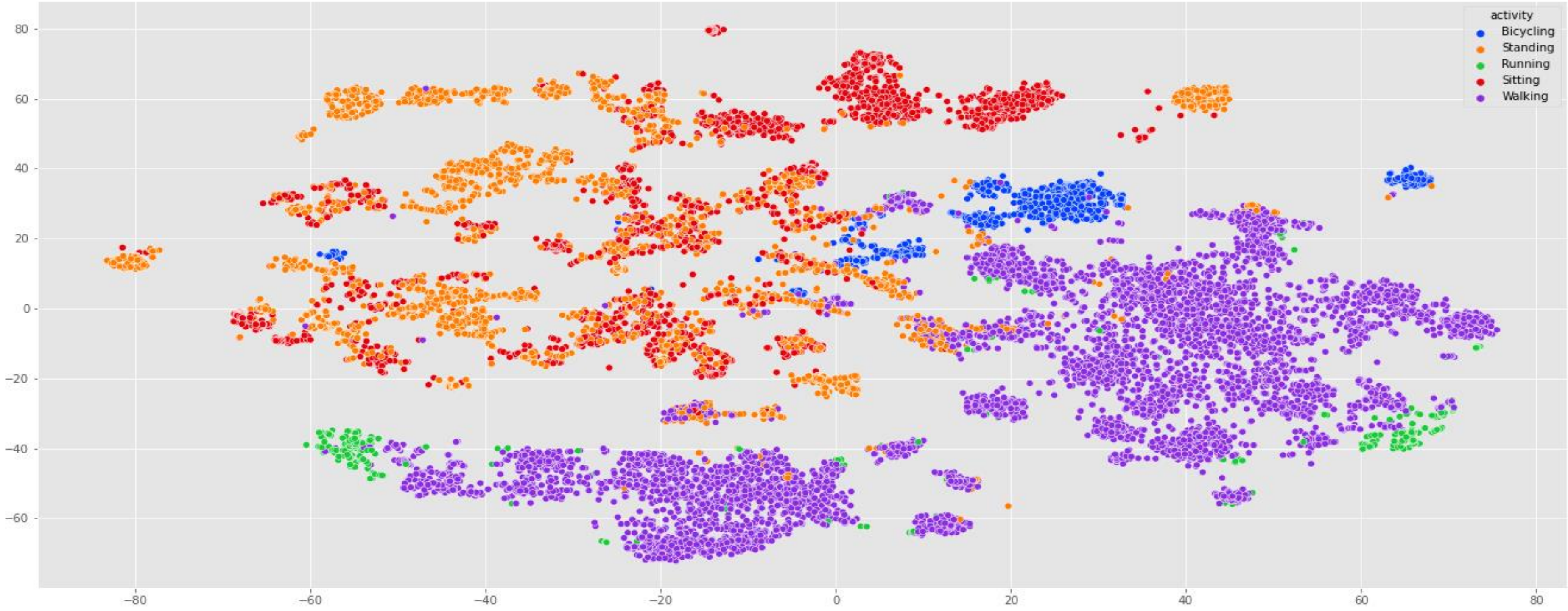
5. Data Preprocessing & Feature Engineering

Features engineering results for boxplot



5. Data Preprocessing & Feature Engineering

Target output engineering and t-sne



6. Models Tuning & Performance

ML models pipeline:

Step 1	Step 2	Step 3	Step 4	Step 5
Make pipeline	GridSearchCV	Model fitting	Cross Val Predict	Accuracy Val Eval

ML models accuracy validation comparison:

Logistic Regression	Decision Tree	Support Vector Machine	K Nearest Neighbors	Random Forest	XGBoost
89.96%	74.31%	92.94%	92.44%	93.63%	95.81%

6. Models Tuning & Performance

XGBoost models tuning results:

```
[ ]  print("Best parameter (CV score=%0.3f):" % XGBoost_grid_fit.best_score_)
     print(XGBoost_grid_fit.best_params_)
```

```
Best parameter (CV score=0.958):
{'xgbclassifier__max_depth': 11}
```

XGBoost classification report:

```
[ ]  print(classification_report(y_lab, y_pred_XGBoost_best))
```

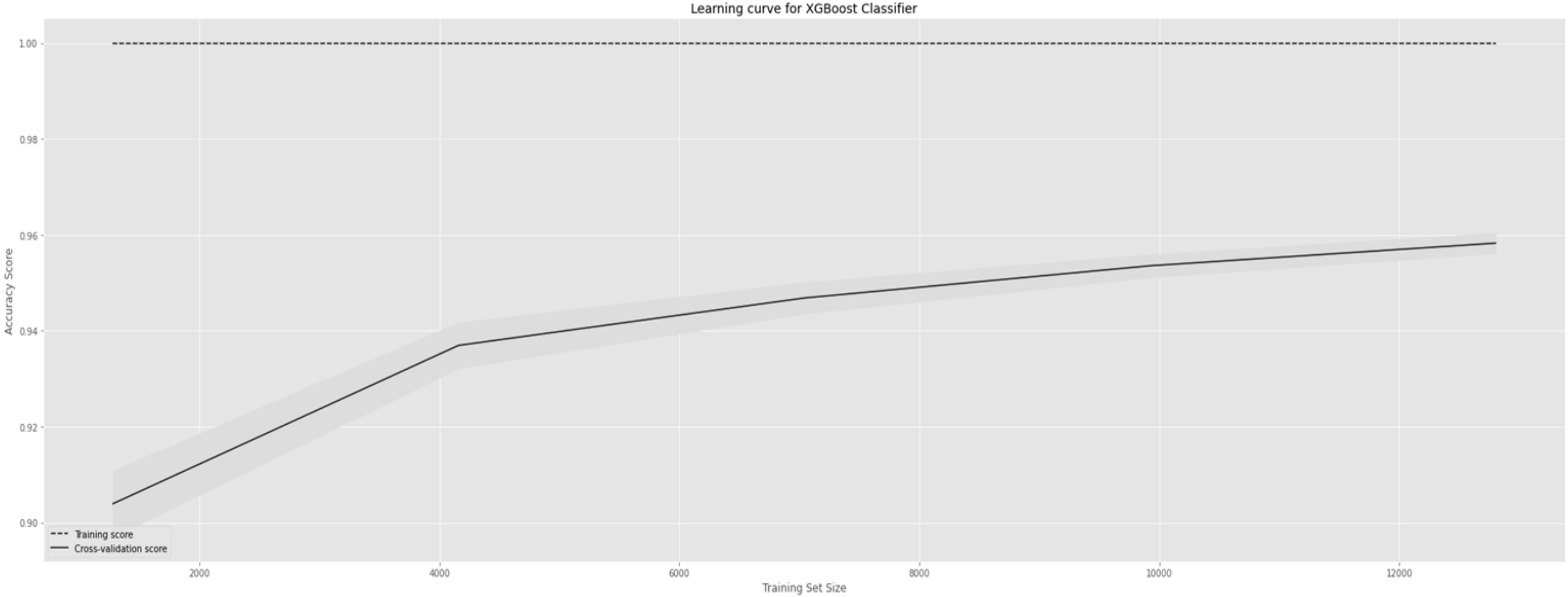
```
              precision    recall  f1-score   support

   Bicycling       1.00      1.00      1.00       908
     Running       0.98      0.94      0.96       493
     Sitting       0.94      0.89      0.92      3228
    Standing       0.91      0.94      0.93      4217
     Walking       0.99      0.99      0.99      7158

    accuracy                           0.96     16004
   macro avg       0.96      0.95      0.96     16004
weighted avg       0.96      0.96      0.96     16004
```

6. Models Tuning & Performance

XGBoost models validation performance:



7. Conclusion: Achieved results

- The best model obtained from the results is XGBoost with ~ 96% accuracy on validation data
- XGBoost combines weak classifiers to build a strong classifiers
- XGBoost works with outliers, non standardized features, collinear features and NaN values
- These results are well comparable to the results obtained on the state of the art (~ 98% for the best models with various databases and methods)
- The solution is built only with 3-axis accelerometer and 3-axis gyroscope based on right foot

8. Conclusion: Recap of the work

Summary

- Leverage SOTA HAR results with features engineering using only one sensors

- Existing works often use complicated neural network architectures and multiple sensors

- The aggregation method (time window) used was inspired by work already done on this subject

- The other models have good accuracy but are less generalized than the XGBoost

9. Next Steps / Roadmap

Prospects for solution improvement

- Use all sensors of the datasets (sensors located at the thighs, shins and quadriceps)
- Explore deep learning models (1D CNN, LSTM)
- Extend the search for the best hyperparameters (fine tunes more parameters)