Danmarks Tekniske Universitet

42186 Model Based Machine Learning

Project Report

Author

Patrik Kucerka - s213425
Mervyn Ho - s226427
Florin Mazilu - s222696

May 29, 2023

Introduction

Abstract

In light of the growing wearable technology market and demand, our aim is to build an Activity Classification model. We used a data set from WISDM Lab, which contains time series data on 36 individuals performing 6 types of activities. These activities include standing, walking, jogging, sitting, ascending and descending stairs. The first model we tried was a binary logistic regression model followed by a multi-classification logistic regression model. While the binary classification model had very high accuracy, 99.9%, in the case of predicting sitting/walking, in the case of predicting walking/jogging the accuracy drops to 75% and for multi-class logistic regression the accuracy drops significantly to 66.5%. The next model we tried was a Gaussian processes (GP) classification model. The final GP model has an 83% accuracy. Therefore, we have decided to use the GP model as our final model.

Background

The wearable technology market is growing due to increased public awareness of health and fitness since the pandemic [2]. Smartwatches and fitness bands with high-tech sensors and rich analytic data are already commonplace in the industry. Activity recognition in these products are a key feature. Therefore, there is a strong importance for activity classification models for this rising market. Some uses of activity classification models include improving smart watches' users' experience and assisting the healthcare industry to track patient activity. Create ML Apple has introduced building Activity Classification Models in 2019 for their products [1]. There have also been research papers published on smartwatch-based activity recognition by Gary M. Weiss, Jessica L. Timko, etc [3]. Since there are multiple use cases for activity recognition, our goal is to build an Activity Classification model based on time series data collected by the accelerometer.

Data selection

The data set was retrieved from WISDM Lab, and contains time series data on 36 individuals performing 6 types of activities. These activities include standing, walking, jogging, sitting, ascending and descending stairs. The data was measured using accelerometers. The table below describes the features of the data set.

Feature	**Description**
user	ID of the accelerometer user
activity	Activity that the user was carrying out
timestamp	Phone's uptime in nanoseconds
x-axis	The acceleration in the x direction as measured by the android phone's accelerometer
y-axis	The acceleration in the y direction as measured by the android phone's accelerometer
z-axis	The acceleration in the z direction as measured by the android phone's accelerometer

Table 1: Table of data set features.

The dataset can be found here: https://www.kaggle.com/datasets/die9origephit/human-activity-recognition

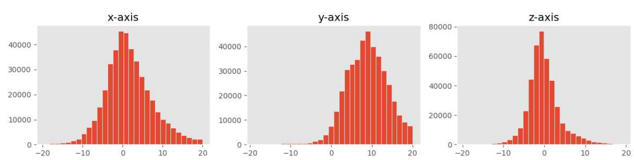
Data description

Firstly, we conducted exploratory data analysis to better understand the data we are working with. We investigated the distribution of activities in the data set and found the following:

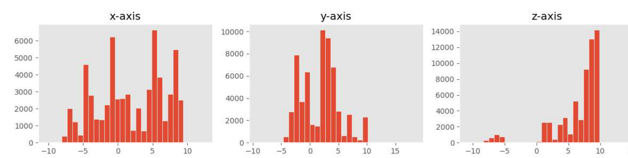
Activity	Percent
Walking	38
Jogging	30
Sitting	5
Standing	4
Upstairs	11
Downstairs	9

Table 2: Table of Activity percentage in data set.

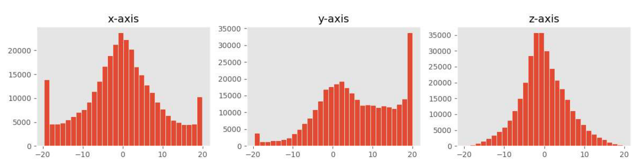
Next, we analysed the descriptive statistics of x-axis, y-axis and z-axis variables for each Activity.



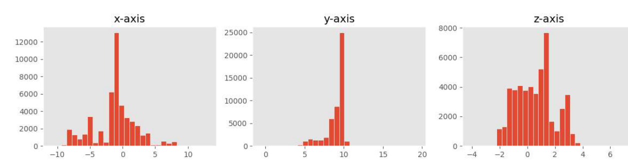
(a) Walking



(b) Sitting



(c) Jogging



(d) Standing

Figure 1: Histogram of the xyz-axis for 4 activities

The tables below are the full descriptive statistics of the x-axis, y-axis and z-axis variables.

Category	Mean (x)	Median (x)	Mode (x)	Standard deviation (x)	Variance (x)
Walking	1.547719	0.990000	-0.460000	5.828952	33.976687
Jogging	-0.227104	-0.230000	-19.610000	9.333826	87.120314
Sitting	1.856270	1.530000	8.200000	4.759028	22.648347
Standing	-1.178269	-1.120000	-1.140000	3.235186	10.466431
Upstairs	0.382177	0.150000	-0.040000	5.501313	30.264444
Downstairs	0.472726	0.080000	-0.230000	4.961663	24.618105

(a) x-axis

Category	Mean (x)	Median (x)	Mode (x)	Standard deviation (x)	Variance (x)
Walking	1.547719	0.990000	-0.460000	5.828952	33.976687
Jogging	-0.227104	-0.230000	-19.610000	9.333826	87.120314
Sitting	1.856270	1.530000	8.200000	4.759028	22.648347
Standing	-1.178269	-1.120000	-1.140000	3.235186	10.466431
Upstairs	0.382177	0.150000	-0.040000	5.501313	30.264444
Downstairs	0.472726	0.080000	-0.230000	4.961663	24.618105

(b) y-axis

Category	Mean (z)	Median (z)	Mode (z)	Standard deviation (z)	Variance (z)
Walking	-0.112612	-0.570000	0.000000	4.020887	16.167532
Jogging	-0.155770	-0.530000	0.000000	5.952950	35.437610
Sitting	6.559716	8.117727	8.240000	3.735591	13.954643
Standing	0.580139	0.650000	1.380000	1.376819	1.895629
Upstairs	0.324257	0.081722	0.000000	3.571793	12.757706
Downstairs	0.685678	0.503953	0.000000	3.710944	13.771104

(c) z-axis

Figure 2: Tables of descriptive statistics for 3 axes

Logistic Regression

This section describes the generative process and PGM for a binary and multi-class logistic regression and their results.

Binary Logistic Regression

Starting with this simple model, we tried to classify just two activities for one person (ID). We tried two different combinations: sitting and walking (which should be fairly easy to classify) and walking and jogging.
The PGM for this task is shown in Figure 3 below:

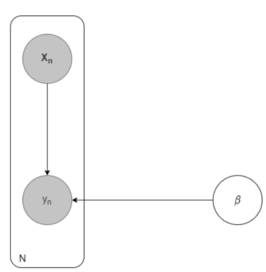


Figure 3: Logistic regression PGM

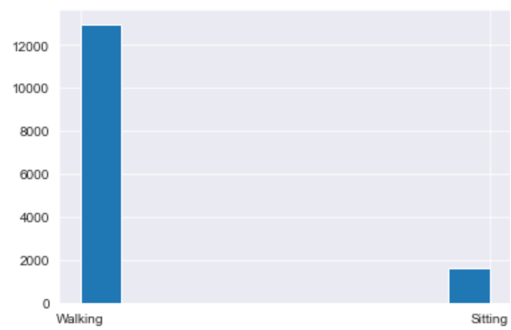
Where $\mathbf{x_n}$ is a vector represented by the x, y and z axes accelerations and $\mathbf{y_n}$ is the predicted activity.
The generative process for the model is:

1. draw coefficients $\beta \sim \mathcal{N}(\beta|0, \lambda I)$

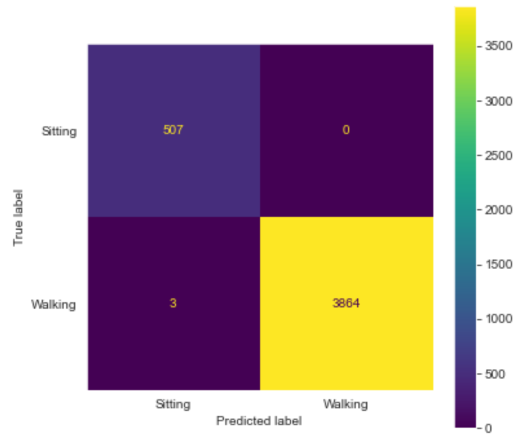
2. for each vector x_n:
   (a) draw class $y_n \sim Bernoulli(y_n | Sigmoid(\beta^\top x_n))$

Results

In the case for sitting and walking, there is a huge imbalance in the data shown in Figure 5a alongside the results, even though the classes are not balanced, the classifier manages to missclassify just three samples.



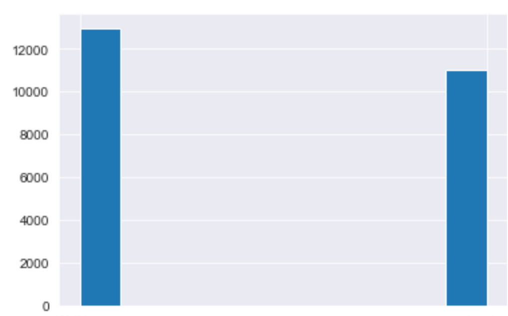
(a) Class imbalance for sitting and walking



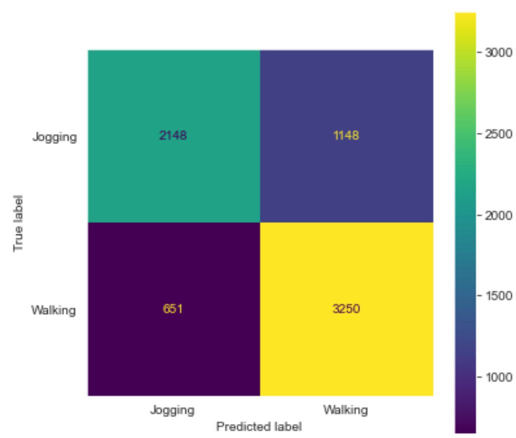
(b) Confusion matrix with 99.93% accuracy

Figure 4

A more complicated case would be where the two activities are closer to each other, such as walking and jogging. In this case the classes are balanced, Figure 5a.



(a) Class imbalance for jogging and walking



(b) Confusion matrix with 75% accuracy

Figure 5

In the first case, the classifier has very high accuracy thanks to the two different activities being so different, in the second case the accuracy drops significantly when the activities start to be similar.

Multi-class logistic regression

In this case, we tried classifying three activities: standing, walking, jogging.
The PGM for this task is shown in Figure 6 below:

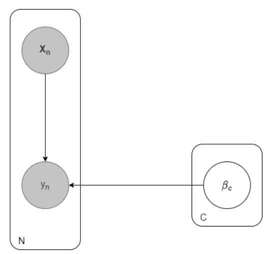


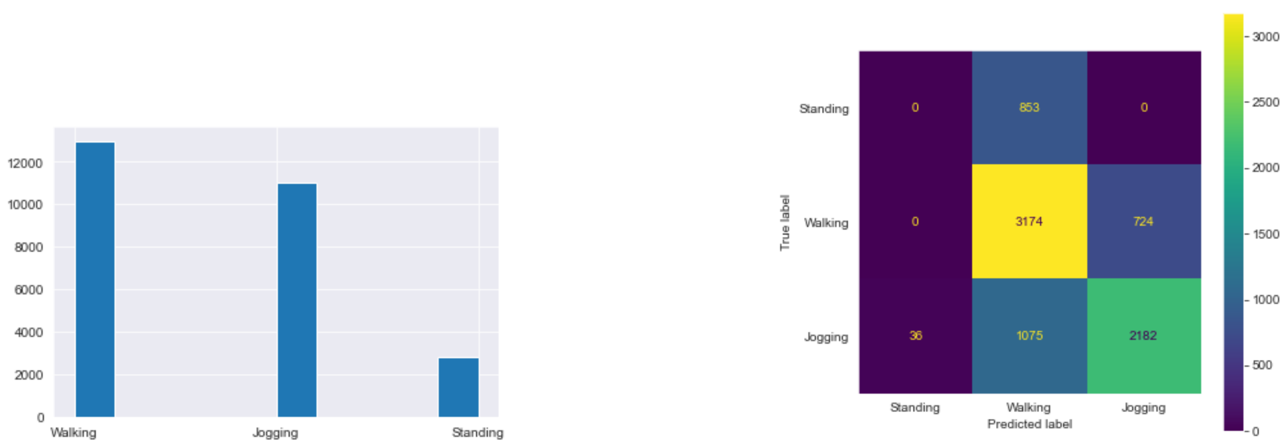
Figure 6: Multi-class logistic regression PGM

The generative process for the model is:

1. for each class $c \in 1, \dots, C$
   (a) draw coefficients $\beta_c \sim \mathcal{N}(\beta_c | 0, \lambda I)$
2. for each vector x_n:
   (a) draw class $y_n \sim Multinomial(y_n | Softmax(x_n, \beta_1, \dots, \beta_C))$

Results

The results alongside class imbalances are presented in the Figure 7 below:



(a) (b)

Figure 7: Class imbalance for standing, jogging and walking (left) and Confusion matrix with 66.5% accuracy (right)

For this task the accuracy of the model drops quite a lot and more interestingly the standing activity is entirely classified as walking.
In the next section, we have tried a different and more complex than the logistic regression, namely Gaussian Process, in the hopes of getting better results.

GP Classification

This section describes the concepts explored and results obtained for a Gaussian Process (GP) classifier.

Down-sampling

The following figure 8 shows down-sampling example. The reasoning for this step is to "encode" the information from multiple measurements into a single data entry, reducing the number of training points (since GPs can be computationally expensive) without losing much information. This decision does not impact the predictions because in real case scenario, user does not need an update 20 times per second (which was the sampling rate). As can be observed, the effect is that the two down-sampled signals are less overlapping compared to the original signal while preserving their means.

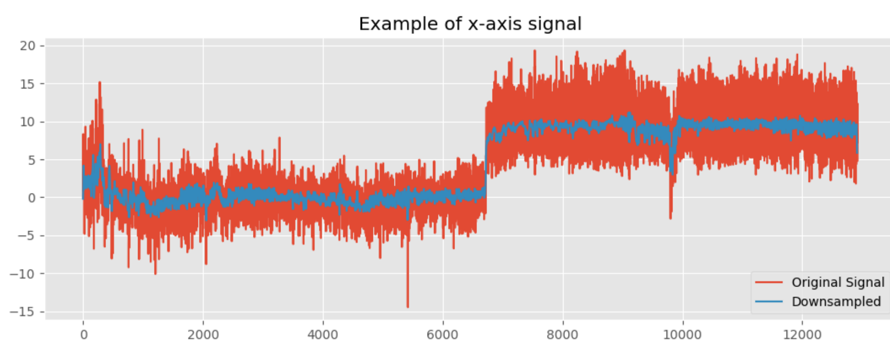


Figure 8: Plot of the down-sampled signal.

Other methods were considered e.g. using spectogram or frequencies of the signals and can be seen in the attached Jupyter file.

Testing Model Suitability

First, the model was tested with only two classes: Sitting and Jogging using only 2 features x and y accelerations. The plot of samples can be seen in the figure 9 below. As can be seen, during Sitting even though the x-axis acceleration is non-zero, it appears to have very small variance while the variance of the Jogging activity is much higher and the data points "surround" the Sitting activity data points.

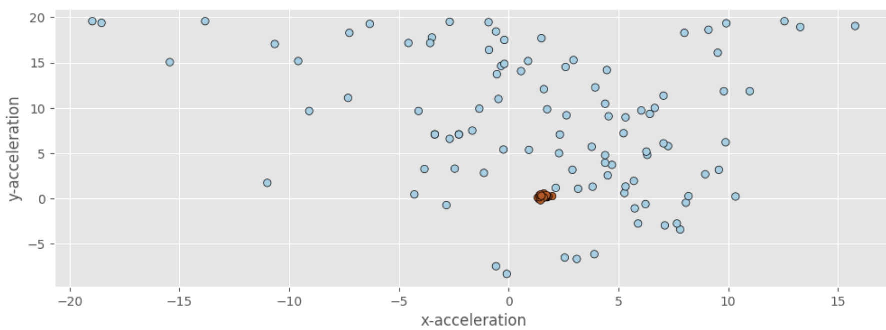


Figure 9: Plot of the two classes: Sitting and Jogging.

The test results can be seen in the figure 10. GP classification model has successfully separated the two classes given with accuracy of 97%. This was also expected as the two classes are easily distinguishable.

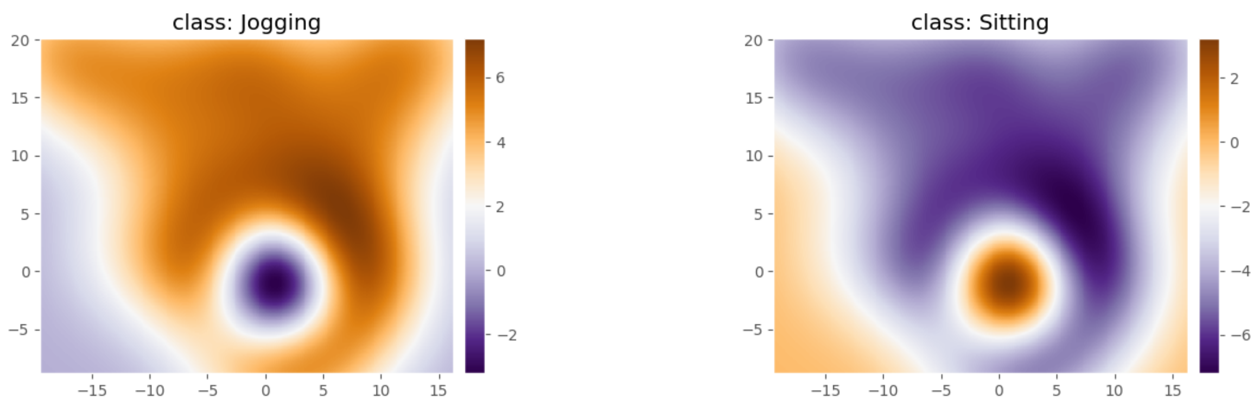


Figure 10: Posterior densities for the two classes.

Model Selection

It is worth mentioning that during the model selection phase, the two suitable kernels were compared: Mattern32 and RBF (Gaussian). They were compared using a more complicated model with 3 classes during 30 experiments. The results can be seen in the following graph 11. It can be concluded that the two kernels perform roughly the same with Matern32 kernel reaching mean accuracy of 91.23% and RBF reaching 90.88%. Even though the difference is minimal, the Matern32 kernel was selected as the results suggest that it performed better.

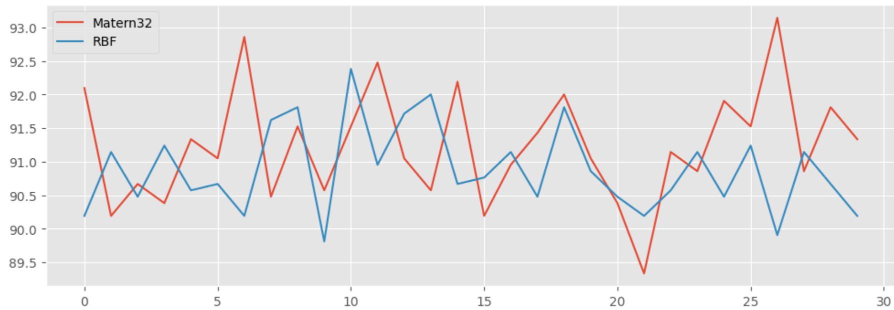
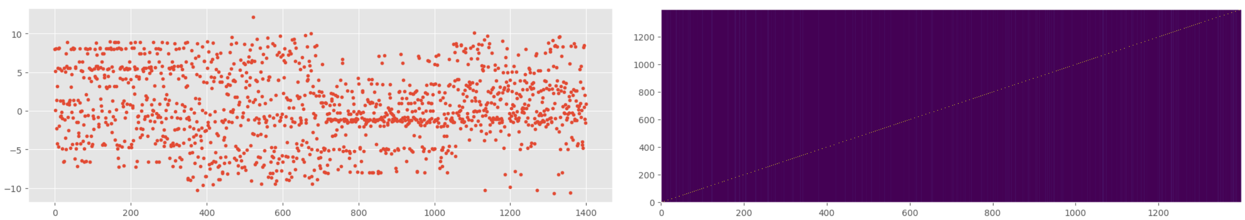


Figure 11: Accuracy reached by Matern32 and RBF kernels.

The reason for testing these two options is that the signal has very low covariance. This can be observed from the following figure 12. The figure on the left 12a shows the acceleration measurements along the x-axis and it can be noticed that the data points are highly uncorrelated. This shows also in the figure 12b which displays the covariance of the measurements using a Gaussian kernel. Since the plot is basically just a diagonal, it implies that the measurements have very low covariance.



(a) Signal measurements. (b) Covariance of the signal using RBF.

Figure 12: Signal's Covariance.

Full model

Model specification

For a Gaussian Process classification task, first step is to model the joint distribution of the targets and latent functions y is p(t,y). This is given by the **likelihood** as a Categorical distribution with K classes:

$$p(t|y) = \prod_{k=1}^{K} \sigma(y)^t$$

and by a **prior** for the latent functions with a covariance function k, which in this case is Matern32:

$$p(y) = \mathcal{GP}(m(x), k(x, x'))$$

Together they give the joint distribution (taking into account all N training points) as:

$$p(t, y) = \prod_{n=1}^{N} (\prod_{k=1}^{K} \sigma(y)^t \mathcal{GP}(m(x), k(x, x')))$$

To compute the posterior p(y|t):

$$p(y|t) \propto \prod_{n=1}^{N} (\prod_{k=1}^{K} \sigma(y)^t \mathcal{GP}(m(x), Matern32(x, x')))$$

it will be necessary to use some approximation method. In this case, a variational inference was chosen for this purpose.

Alternatively the model can be displayed as a PGM as displayed in the figure 13. Where x and t are observable inputs (vector of acceleration measurements) and output (activity category), respectively. y represents the output of a latent function that is not directly observable and assumes a linear relationship between some parameters θ and basis functions ϕ.

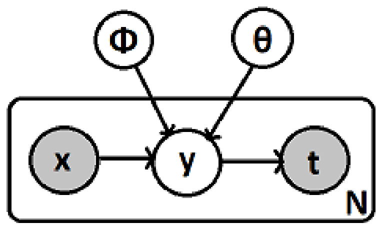


Figure 13: PGM for the GP Classifier.

Training

Next, the model was trained for 4 classes: Sitting, Jogging, Standing and Walking. The model was trained using Matern32 kernel and the training loss function can be observed in the figure 14 below. As can be seen in the first 400 iterations, the error was rapidly decreasing and reached steady state error at around 770.

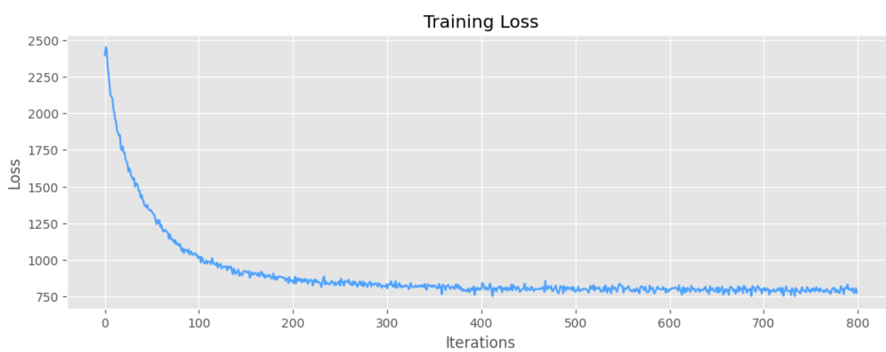


Figure 14: Training Loss.

When the model was tested, it reached **accuracy** of **83%** which is much better then a random guess. The results were visualized with a confusion matrix that can be seen in the figure 15 below. It can be noticed, that interestingly the most challenging part is distinguishing between the standing activity and walking with slightly high occurrence then the expected issue to distinguish between walking and jogging (this was expected to be problematic as the the two activities are very alike).

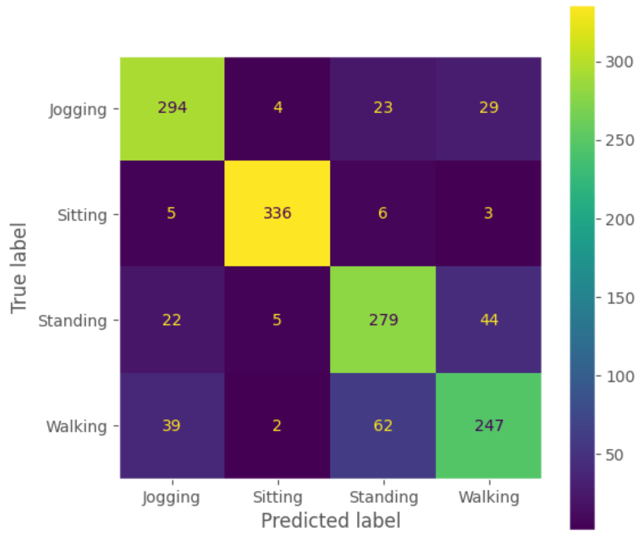


Figure 15: GP Confusion Matrix.

Conclusion for GP Classification

During the experiments, two kernels were chosen based on the data covariance and explored. Both perform equally well but Matern32 kernel was selected with a slightly higher accuracy results. The GP classification model managed to distinguish between the 4 classes given (Walking, Jogging, Sitting and Standing), with an accuracy of 83 %. This means that the model performs much better then a random guess (which would be 25%) and can be potentially used for a scuh task. In the future, the model could be further improved by collecting more information e.g. GPS signal to better distinguish between the walking class and jogging class as the persons velocity would provide information not captured by the acceleration information from ones wrist.

References

[1] Apple Inc. Building activity classification models in create ml, Jun 2019.

[2] Sang M Lee and DonHee Lee. Healthcare wearable devices: an analysis of key factors for continuous use intention. *Service Business*, 14(4):503–531, 2020.

[3] Gary M Weiss, Jessica L Timko, Catherine M Gallagher, Kenichi Yoneda, and Andrew J Schreiber. Smartwatch-based activity recognition: A machine learning approach. In *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 426–429. IEEE, 2016.