**MSc in Business Analytics**

****

**Big Data Content Analytics**

**Neural Networks Assignment**

**Aggeliki Siampli - AM: ΒΑΡΤ1727**

**Dimitrios Thomas Damtsias - AM:** **BAPT1705**

**Lia Yfanti - AM:** **BAPT1707**

**Manos Nikolopoulos - AM:** **BAPT1720**

Table of Contents

[1. Project Description 3](#_Toc524560840)

[What are we trying to accomplish and why would this be important to anyone: 3](#_Toc524560841)

[Is the task feasible business-wise? 3](#_Toc524560842)

[Business workflows: 4](#_Toc524560843)

[Approach-Awareness 4](#_Toc524560844)

[Daily Usage 4](#_Toc524560845)

[2. Mission: 4](#_Toc524560846)

[3. Data to be used 4](#_Toc524560847)

[Pre-processing method: 6](#_Toc524560848)

[Information about our data: 6](#_Toc524560849)

[Visualization of our data: 7](#_Toc524560850)

[4. Methodology – Results 9](#_Toc524560851)

[5. Members/Roles 13](#_Toc524560852)

[6. Bibliography 14](#_Toc524560853)

[7. TimePlan 14](#_Toc524560854)

[8. Notes / Comments 15](#_Toc524560855)

[9. Contact Person 15](#_Toc524560856)

[Figure 1 Frequency per Activity 8](#_Toc524560857)

[Figure 2 Activity per Subject 8](#_Toc524560858)

[Figure 3 Dispersion of Activities 9](#_Toc524560859)

[Figure 4 Our MLP model 11](#_Toc524560860)

[Figure 5 Plot indicating the evolution of loss and accuracy over epochs 12](#_Toc524560861)

[Figure 6 Confusion Matrix Plot 13](#_Toc524560862)

# Project Description

## What are we trying to accomplish and why would this be important to anyone:

With the concept of fitness and well-being steadily growing around the world, more and more companies that aim to capitalize on this demand are constantly being formed. Companies that offer a variety of products (healthy food, sports goods, gym equipment, well-being consulting services, etc.), that are joining the already established line of companies that aim to gain from an overall healthy way of living (insurance companies, state and private hospitals, sports events sponsors, etc.)

Our goal is to provide them with a marketing tool that they can offer to potential clients to increase their brand awareness and profitability. An application that will detect the current activity of a user and will provide incentives for him/her to change his daily routine so as to include more and more activities that have to do with some form of physical activity.

As a marketing tool this app could:

* Provide points to the user for whenever he/she walks to a destination instead of driving
* Provide points to the user for whenever he/she takes the stairs instead of taking the elevator
* Provide points to the user for whenever he/she completes a well-rested sleep

These points (that are merely an example) could be used by a company to provide awards/benefits to the users (gifts, discount vouchers, etc.) for whenever they engage in a physical activity.

As a recording tool this app could:

* Track the daily fitness activity of a user (how many miles he has walked, how long he is sitting down instead of standing up, etc)
* Keep records that the user could send to a doctor/hospital/insurance company, for feedback so as to get a potentially better diagnosis/treatment/insurance contract

## Is the task feasible business-wise?

Marketing tools that offer customer reward schemes and track customer behavior are already present in the business world and are already widely used. Our suggestion involves the use of smartphones, which is ideal since everyone has a smartphone nowadays and all smartphones have the basic architecture needed for our task (GPS, gyroscope, compass, G-force tracking, accelerometer, maps, etc.). They are all there, running in the background, waiting for someone to take advantage of their capabilities.

A lot of companies have already implemented a number of solutions (i.e. software that measure how many steps you have taken during the day, software that track driving styles to promote safe driving, etc.). All these success stories are there to show us that a simple solution with minimum investment could be offered via an existing structure (smartphone, app, smartwatch, etc.) with meaningful impact.

So in a nutshell, we are talking about the use of a piece of software that could detect human activity and would simply need to be connected to some hardware, so quite feasible business-wise and with reasonable costs.

## Business workflows:

To use this tool properly, a company would have to embed it in their customer lifetime cycle. From the moment they first approach him as a potential lead, up until the time they try to offer him incentives to prevent him from churning.

## Approach-Awareness

This tool could be part of the company’s USPs (unique selling points). An incentive to approach and convert customers. It would have to be part of the company’s main customer platform (app, website customer area, etc.) and present at all times

## Daily Usage

The tool must run at all times and record everything with limited internet and battery usage and of course no costs for the customer. The data can be collected and sent back when the customer is asleep at night and at his home, connected to his wi-fi. Daily analysis must be present too, so the company needs to be able to process large data and use rules to trigger certain actions (i.e. give away a voucher because the customer reached a certain milestone). The company does not necessarily need to store all the data, just the information they need. The use of streaming analytics could be a must.

# Mission:

Our mission is to train a Neural Network that will predict/detect human activity through the daily use of smartphones. This is a task that has been tackled before with quite a few examples running in the academic circles. However, we are talking about a business application of such a network.

So, our goal is to train a Neural Network with data for which we know the users activity and use that trained network along with a device’s built-in features for prediction. The Network will collect data from the device’s gyroscope, accelerometer, etc. and try to predict the user’s activity based on specific classes it has used for training before. Our mission is basically a classification problem that will run continuously on several devices.

# Data to be used

We came across a dataset[[1]](#footnote-1) online that was built from the recordings of 30 study participants performing activities of daily living while carrying a waist-mounted smartphone with an embedded inertial sensor.

This dataset has 10.300 records of 563 variables. These variables were produced from the smartphones embedded accelerometer and gyroscope and provide information regarding for example: Triaxial body acceleration information (mean, std, max, mad, entropy, etc.), Triaxial gravity acceleration information (mean, std, max, mad, entropy, etc.), Triaxial angular information, Body jerk information, etc.

At the end, the dataset has 6 labels: Standing, laying, sitting, walking, walking downstairs, and walking upstairs. These labels will serve as our predicted classes.

The following table shows the distinct categories for which measures have been taken. Notice that the last two categories refer to the subject (number identifying the person that produced the data in the row) and the activity. Activity takes the following values: STANDING, SITTING, LAYING, WALKING, WALKING\_DOWNSTAIRS, WALKING\_UPSTAIRS.

| Measure category | Notes |
| --- | --- |
| tBodyAcc | These measures are taken for X, Y, Z axes |
| tGravityAcc |
| tBodyAccJerk |
| tBodyGyro |
| tBodyGyroJerk |
| tBodyAccMag |
| tGravityAccMag |
| tBodyAccJerkMag |
| tBodyGyroMag |
| tBodyGyroJerkMag |
| fBodyAcc |
| fBodyAccJerk |
| fBodyGyro |
| fBodyAccMag |
| fBodyBodyAccJerkMag |
| fBodyBodyGyroMag |
| fBodyBodyGyroJerkMag |
| angle(tBodyAccMean,gravity) |
| angle(tBodyAccJerkMean),gravityMean) |
| angle(tBodyGyroMean,gravityMean) |
| angle(tBodyGyroJerkMean,gravityMean) |
| angle(X,gravityMean) |
| angle(Y,gravityMean) |
| angle(Z,gravityMean) |
| subject | Unique number for each participant |
| Activity | STANDING, SITTING, LAYING, WALKING, WALKING\_DOWNSTAIRS, WALKING\_UPSTAIRS |

Table 1 Measure's Categories

## Pre-processing method:

We normalized the data we used in our analysis using the following function for our Train and Test dataset:

a) (Train\_data – Mean\_of\_trained\_data) / Standard\_Deviation\_of\_Train\_Data

b) (Test\_data – Mean\_of\_test\_data) / Standard\_Deviation\_of\_Test\_Data

## Information about our data:

In this section we are going to present some numeric and formation information of the dataset we used to our analysis:

a) Here are the number of records used for the training and testing of our network, as well as the number of variables taking place in each procedure:

Information extracted from our python program, in the form:

“Train/Test\_set: (number of records, number of variables)”

**Shape Train: (7352, 564)**

**Shape Test: (2947, 564)**

b) Here we have a representation of how our data are stored:

Each variable label contains the name of the metric (e.g. “tBodyAcc”), the type of metric (e.g. “mean()”), and the geospatial axis in which is recorded (e.g. “X”):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tBodyAcc-mean()-X | tBodyAcc-mean()-Y | ... | Activity | Data |
| 0 | 0.288585 | -0.020294 | ... | STANDING | Train |
| 1 | 0.278419 | -0.016411 | ... | STANDING | Train |
| 2 | 0.279653 | -0.019467 | ... | STANDING | Train |
| 3 | 0.279174 | -0.026201 | ... | STANDING | Train |
| 4 | 0.276629 | -0.016570 | ... | STANDING | Train |

Table 2 Sample of our data

Regarding the type of the variables used, we have:

a) 561 variables of type “float” and

b) 2 variables of type “Object”, which are the “Activity” information (the output move type of the record) and the “Data” information which inform us about the set that this recordincluded (Train or Test).

c) Here we show the main features from which our dataset consists of and the number of different variables that exist in it and relate to these main categories. For example, the category “tBodyAcc” has 79 variables in this dataset:

|  |  |
| --- | --- |
|  | Count |
| fBodyAcc | 79 |
| fBodyGyro | 79 |
| fBodyAccJerk | 79 |
| tGravityAcc | 40 |
| tBodyAcc | 40 |
| tBodyGyroJerk | 40 |
| tBodyGyro | 40 |
| tBodyAccJerk | 40 |
| tBodyAccMag | 13 |
| tGravityAccMag | 13 |
| tBodyAccJerkMag | 13 |
| tBodyGyroMag | 13 |
| tBodyGyroJerkMag | 13 |
| fBodyAccMag | 13 |
| fBodyBodyAccJerkMag | 13 |
| fBodyBodyGyroMag | 13 |
| fBodyBodyGyroJerkMag | 13 |
| angle | 7 |
| subject | 1 |
| Data | 1 |

Table 3 Indicating how many variables there are per measure category

## Visualization of our data:

In this section we are going to show some visualizations of our data:

a) Here we have the total number of Train Set records per “Activity”:

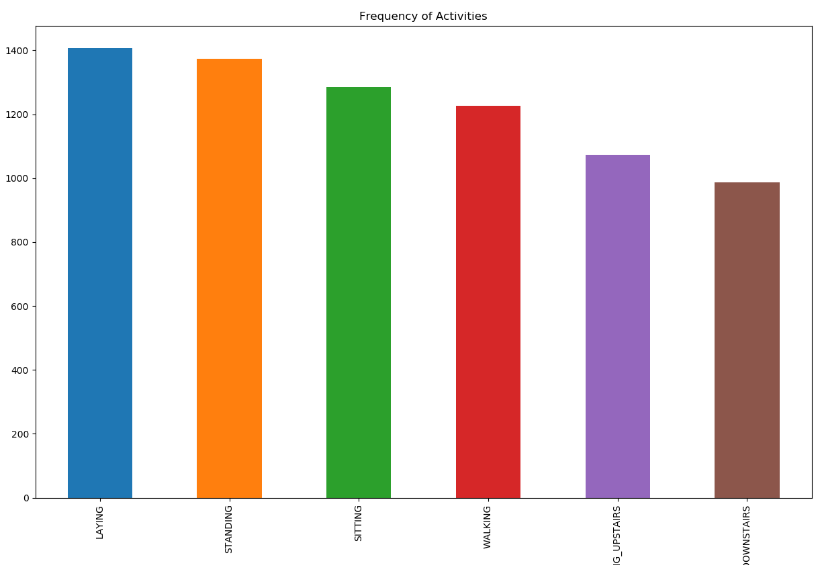


Figure 1 Frequency per Activity

As we can see the records are distributed equally enough to all Activity types, with more observations on “LAYING” Activity and less on “WALKING\_DOWNSTAIRS” Activity.

b) Here we present the amount of action of each subject (individual participant) per each of the Activities he/she made:

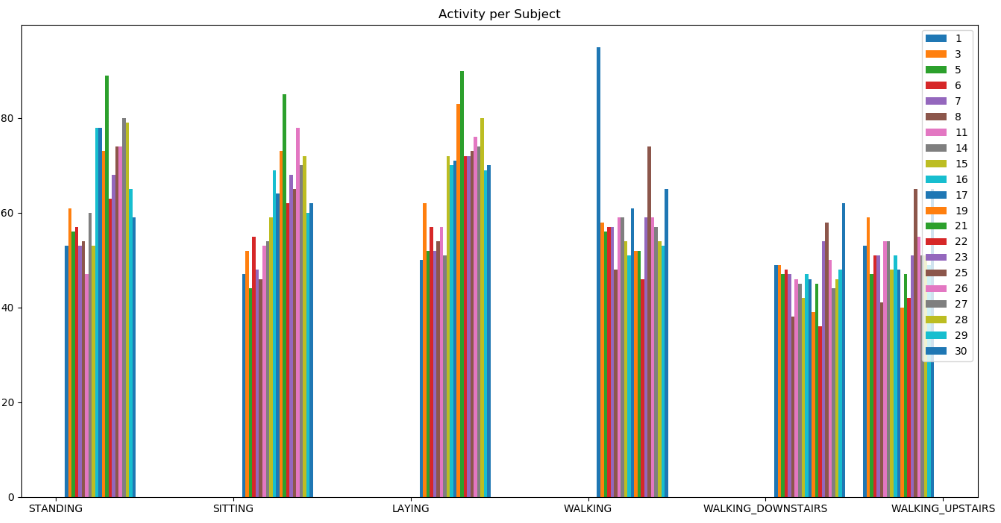


Figure 2 Activity per Subject

As we can see for each activity, each participant has made the same Activity almost as many times as the others.

c) The following plot is descriptive enough regarding the dispersion of Activities:

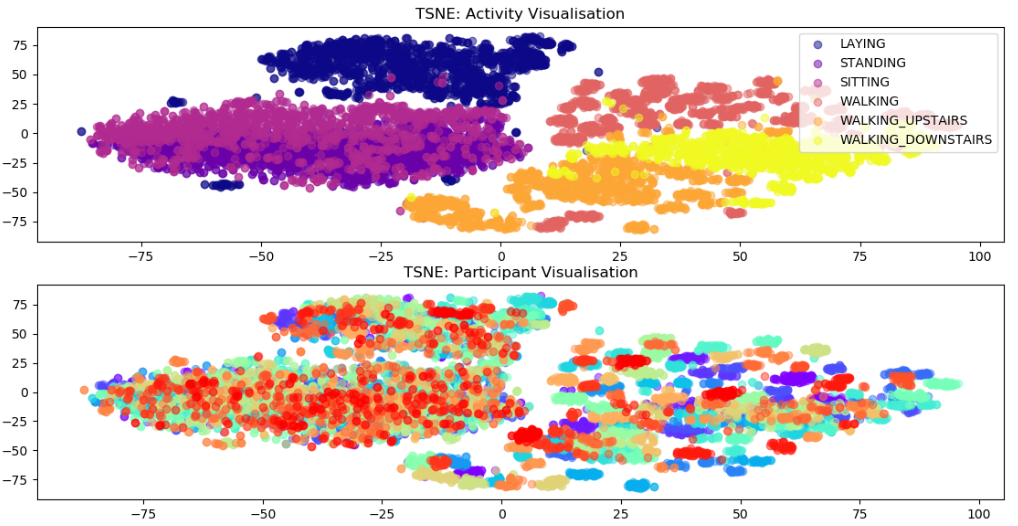


Figure 3 Dispersion of Activities

As we we can see from the above visualization “TSNE: Activity Visualization”, the Activities are quite separable. That means that each specific Activity has some unique characteristics which are accurately enough captured by the smartphones during the experiment phase.

Regarding the second plot with name “TSNE: Participant Visualization”, where each participant represented with different color, we can see that they have pretty much custom moving patterns. That’s why on the right formatted cluster (which represents the 3 moving activities: WALKING, WALKING\_UPSTAIRS, WALKING\_DOWNSTAIRS), we can clearly see each participants action.

On the other hand, on the left side of the plot, which represents the “moveless” Activities (SITTING, STANDING, LAYING), we can observe that the behavior of participants is similar to each other. This is normal because all persons does the same moves while they are sitting, laying or standing.

# Methodology – Results

In this assignment, we tried to solve the HAR problem by adapting one particular deep learning model - the multilayer perceptron neural network (MLP ).In general, MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

More specific now, the method we have implemented, adopts a deep multilayer perceptron neural network (MLP ) to automate feature learning from the raw inputs in a systematic way.

Through the deep architecture, the learned features are deemed as the higher level abstract representation of low level raw time series signals. By leveraging the labelled information via supervised learning, the learned features are endowed with more discriminative power. Unified in one model, feature learning and classification are mutually enhanced. All these unique advantages of the MLP make it out-perform in HAR problem resolution.

More specific, we started by making a feed-forward neural network with three fully connected layers - two visible (input and output layers) and one hidden- combined with two dropout layers implemented on the input layer and on the hidden layer.

We decided to make use of dropout layers in order to deploy a network that is capable of better generalization and avoid at the same time overfitting since we have a large number of input variables (561).

Each layer is composed of 64 neurons apart from the output layer that has 6 which are the unique classes of our data (WALKING, WALKING\_UP, WALKING\_DOWN, SITTING, STANDING, LAYING).

Each layer represents a mapping of a tensor (defined as multidimensional matrix) to another.

* may have arbitrary shapes
* Each layer is configured using a set of parameters

Training of a neural network organized into layers is achieved by backpropagation. For each layer which is built using differentiable functions, the backpropagation algorithm is applied to tune the layer parameters algorithmically based on the available training data.

Each dense layer is specified by:

* Input dimensionality: the input tensor is of the shape.
* Output dimensionality: the output tensor is of the shape.
* An activation function that reacts to the n activation signals.

A dense layer is defined as:

The input of the model is all the columns of the HAR dataset apart from the activity column, which is actually the output class in our model, and the subject column which is irrelevant to our classification problem. Summarizing, we have 561 different measurements delivered by the accelerometer and gyroscope of a smartphone that consist of captures of 3-axial linear acceleration and 3-axial angular velocity.

Before using the data as input to our model we had to normalize the values in order to have the same range of values for each of the input columns. This would guarantee us stable convergence of weight and biases and as we observed later it gave a slight increase in our model’s accuracy in comparison with the accuracy we had without normalized data.

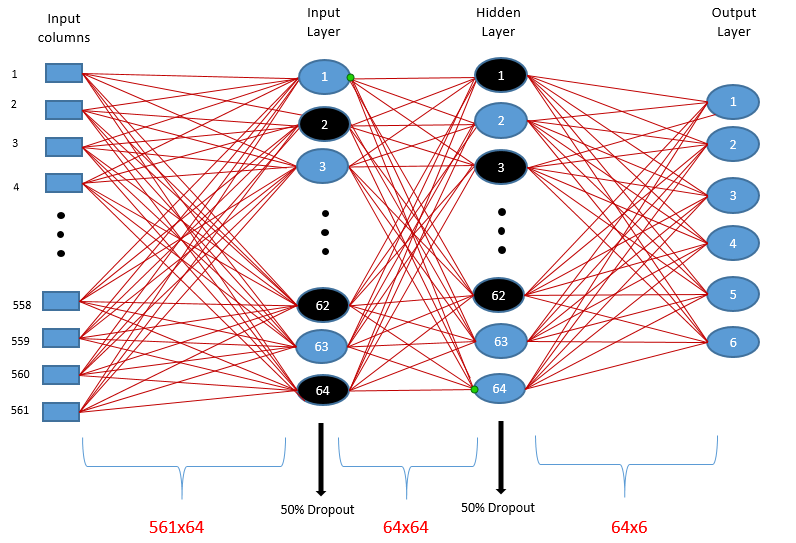


Figure 4 Our MLP model

Regarding the trainable parameters of our model, we have 35968 parameters in the input layer which is actually the number of input variables (561) multiplied by the number of neurons (64), in the second layer we have 4160 parameters and in the output layer we have 390 which is the number of neurons multiplied by the output of the previous layer. At the end, we have a total of 40518 trainable parameters.

We set the number of epochs to 40 but we saw that our model was starting to converge at 33 epochs and it didn't make any additional progress after that, so we decided to decrease the number of epochs to 30.

On the fitting process, regarding the batch size, we knew the most common used batch sizes are 64, 128, 256 and generally powers of 2. We started with setting it to 128.The accuracy of our model was about 92% but the duration of training time was about a few seconds. Since the training process was relatively fast, it gave us the opportunity to increase the batch size to 256.The results were much better since we achieve 95% accuracy with hardly any consequences in time duration and GPU memory usage.

After that we had to decide which activation function we are going to use for each layer. The activation function we used rectified linear unit (ReLU) and took advantage of sparsity and reduced likelihood of vanishing gradient.

The (leaky) Rectified Linear Unit (ReLU) activation function is called:

and behaves as follows:

If max\_value is defined, the result is truncated to this value.

On the output layer we used softmax because we have multi classification problem. The softmax activation normalizes data along the specified axis.

About the optimizer, we started our tests with stochastic gradient descend (SGD) with learning rate at 0.01 and momentum at 0.9 but then we found out that Adam optimizer with reduced learning rate at 0.001 gave us better results since it added proximately 1% to our accuracy.

Next step was the evaluation of our trained model. To evaluate it, we computed the training and validation cross-entropy loss and the train and validation accuracy per epoch. As it can been seen from the plot below, the accuracy is increased over time and the loss is decreased.

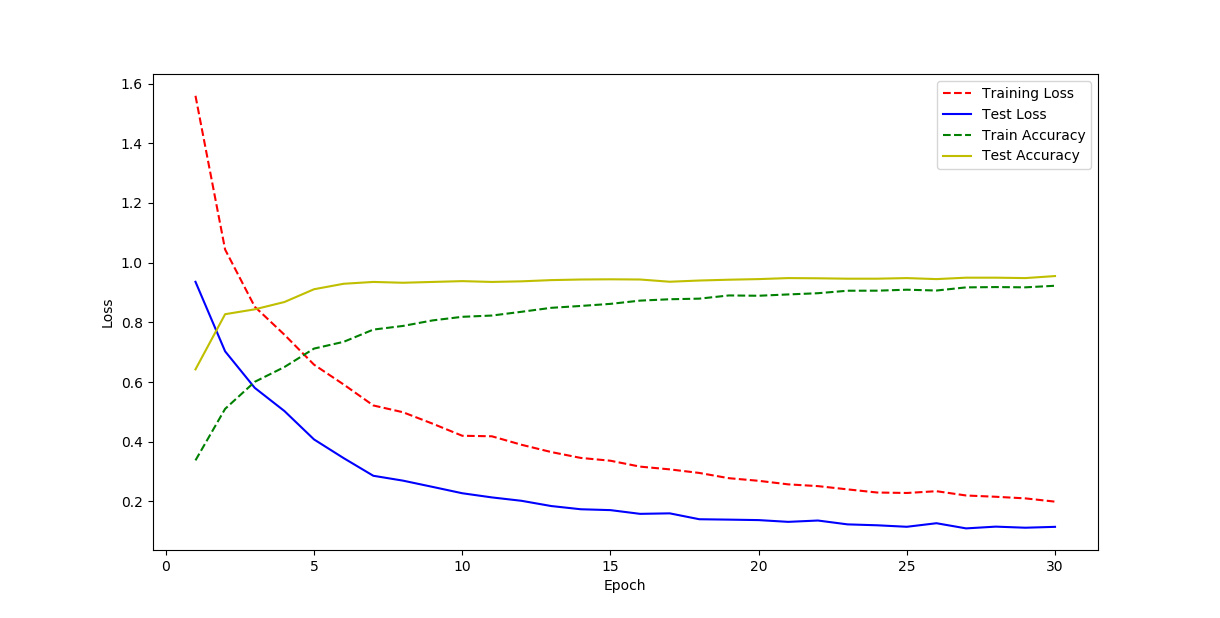


Figure 5 Plot indicating the evolution of loss and accuracy over epochs

One other point we observed from the above plot is that train loss starts much higher from validation loss. This happens because the training loss is the average of the losses over each batch of training data. Because the model is changing over time, the loss over the first batches of an epoch is generally higher than over the last batches. On the other hand, the testing loss for an epoch is computed using the model as it is at the end of the epoch, resulting in a lower loss.

The same observation was made between validation and training accuracy. In this case, we believe, that is happening because of the dropout we have added in the layers. In the training process 50% of the features are set to zero. When testing, the dropout is not taking place so all the features are used and this leads to higher validation accuracy.

To visualize the cases where our model did not predict right we created a confusion matrix and a confusion matrix plot as can been seen below.

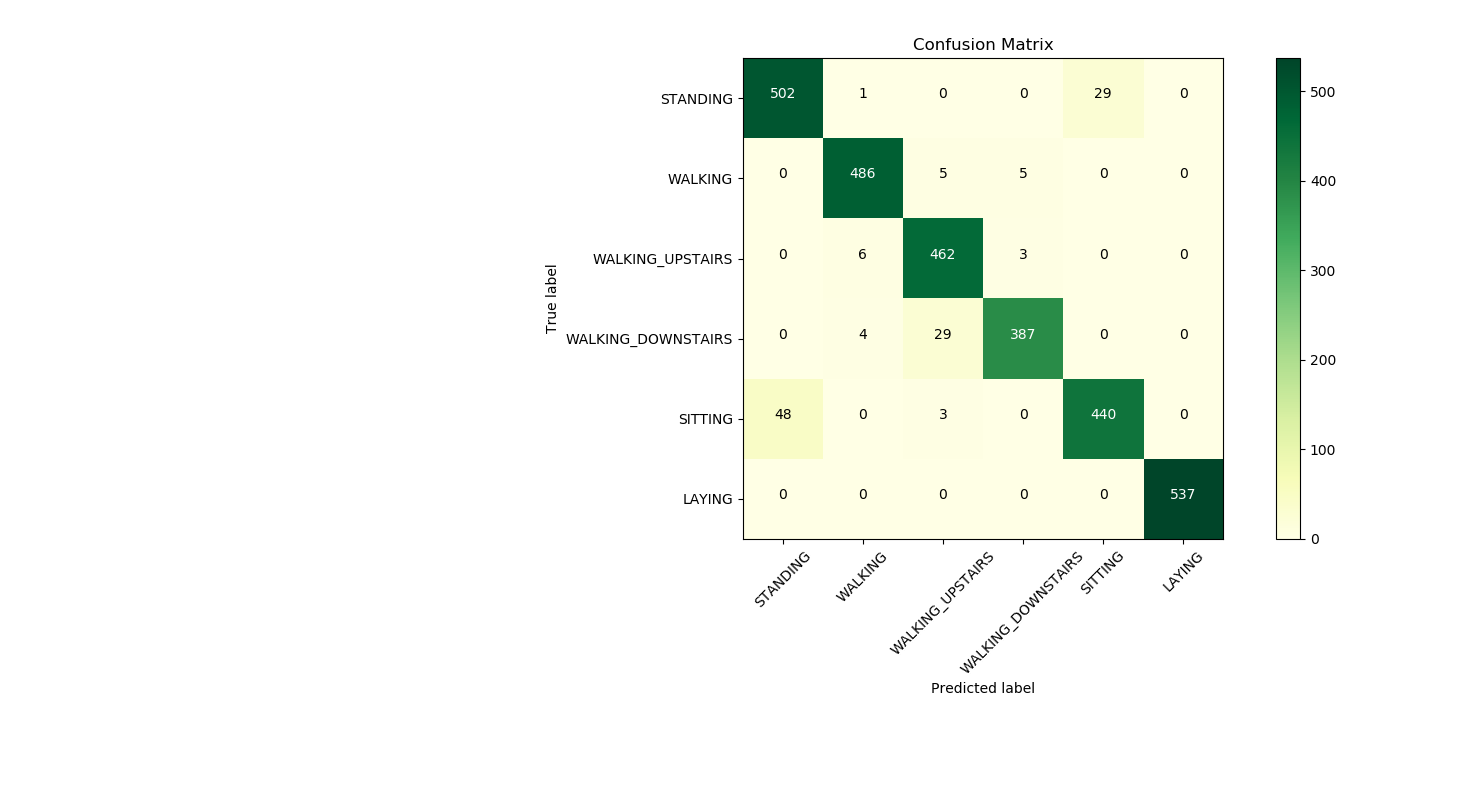


Figure 6 Confusion Matrix Plot

It is obvious that most of the faulty decisions are between STANDING and SITTING which makes sense since the experiment was done with the use of a smartphone wearied on the waist of the subjects. So the differences between the axis in standing and sitting were not significant

All of the above operations were developed and executed on a Intel Core I7-4700 desktop with a CPU at 2.40GHZ, two RAMs of 8GB each that runs Windows 10 64-bit operating system.

# Members/Roles

**The team:**

· Aggeliki Siampli (programmer)

**Brief Background:**

Software Engineer. Graduate of Digital Systems at University of Piraeus. Familiar with Java and Python.

· Dimitris Damtsias (data engineer)

**Brief Background:**

CRM Administrator. Graduate of Management Science and Technology of AUEB. Familiar with data management, ETL process and visualization of data.

· Lia Yfanti (data scientist)

**Brief Background:**

Computer Scientist Familiar with : SAP Software (ERP Software System), QlikView program for reporting and annual IT budget management

· Manos Nikolopoulos (business analyst)

**Brief Background:**

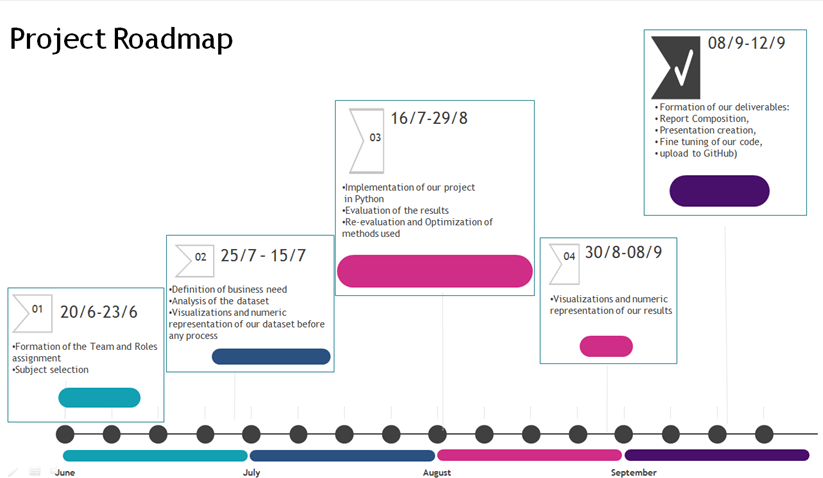
Mathematician and an experienced Business Analyst in the insurance sector

# Bibliography

**References links:**

* http://www.mdpi.com/1424-8220/16/1/115
* https://dl.acm.org/citation.cfm?doid=2733373.2806333
* https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones
* https://dl.acm.org/citation.cfm?doid=1964897.1964918
* https://www.sciencedirect.com/science/article/pii/S1877050915000320

# TimePlan



# Notes / Comments

The difficult part was to decide which type of Neural Network we should deploy in order to solve the problem with the most effective way. Because there is no clear rule of thumb everything must be tested to see which works best.

Regarding teamwork experience, different team members contribute different perspectives, and the synergy between team members can produce creative and productive results. Through conversations and controversies about each aspect of the project, we end up with a solution that make our project functional and output good quality results.

# Contact Person

Manos Nikolopoulos – Email: manosnikolop@gmail.com

1. Full information regarding the dataset can be found in:

   https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones# [↑](#footnote-ref-1)