Owen Agius, Sean Farrugia, Francesca Maria Mizzi

owen.agius.19@um.edu.mt sean.farrugia.19@um.edu.mt francesca.mizzi@um.edu.mt

This project is concerned with classifying human activity (e.g., walking, standing, etc...) from sensor data captured using a smartphone. In this task, students will be expected to study and apply various machine learning techniques to build a classification model from training and validation data sets.

Human activity recognition

* Group Assigned Practical Task

Table of Contents

[Chapter 1: Introduction 3](#_Toc71143271)

[Chapter 2: Design Overview 4](#_Toc71143272)

[2.1: Supervised Machine Learning 4](#_Toc71143273)

[2.1.1: Raw Data Pre-processing 4](#_Toc71143274)

[2.1.2: Data Analysis 4](#_Toc71143275)

[2.1.3: Predictions of Data 5](#_Toc71143276)

[Chapter 3: Technical Overview 6](#_Toc71143277)

[3.1: Support Vector Machines 6](#_Toc71143278)

[3.1.1: Radial Kernel 6](#_Toc71143279)

[3.2: K Nearest Neighbours 6](#_Toc71143280)

[3.2.1: Ball Trees 6](#_Toc71143281)

[3.2.2: KD Trees 7](#_Toc71143282)

[3.3: Decision Trees 7](#_Toc71143283)

[3.4: Logistic Regression 7](#_Toc71143284)

[Chapter 4: Code Overview 8](#_Toc71143285)

[Chapter 5: Testing and Results 9](#_Toc71143286)

[Chapter 6: Conclusion 10](#_Toc71143287)

[Chapter 7: References 11](#_Toc71143288)

# Chapter 1: Introduction

# Chapter 2: Design Overview

The task of identifying human activity can be described as a classification problem. A classification problem has a discrete value as its output. For example, “sitting” and “walking upstairs” are discrete. There is no middle ground. Generally, Supervised Machine Learning algorithms are used in order to tackle classification or regression problems.

## 2.1: Supervised Machine Learning

Algorithms which adopt Supervised Machine Learning are designed to learn by example. While training a supervised learning algorithm, the data within the training data-set will contain inputs with fields determining their correct output.

After training, the algorithm will take new unseen data and will attempt to determine the correct output for the new inputs based on the prior training data. Therefore, it can be generalized that a supervised learning model's aim is to predict the correct cluster or label for newly presented data.

The Human Activity data was able to be recognized by undergoing a three-stage procedure. The first step entails for the original data collected to be first transformed into an earth plane from a bodily plane rendering the data usable. Secondly, the data is analysed and visualized to assist in determining the best approach for the third and final step, and assisting in determining any issues or errors in clustering. Finally, through the third step, the data set is split into training and testing sets. Using supervised machine learning algorithms, a prediction is rendered.

### 2.1.1: Raw Data Pre-Processing

### 2.1.2: Data Analysis

After checking both the training and testing data sets for duplicate values and NaN or Null values, the aim of the Data Analysis notebook is to properly cluster and visualize the optimized clusters on the training data set. The clustering of the data set is optimized using the t-SNE algorithm on the data.

An example of the clustering can be found below. It can be observed that the moving activities are fairly distinct from one another, besides a few outliers. Meanwhile, the stationary activities can easily be confused for each other.

Since the data is collected from inertia sensors and is based on movement, the below clustering is expected.

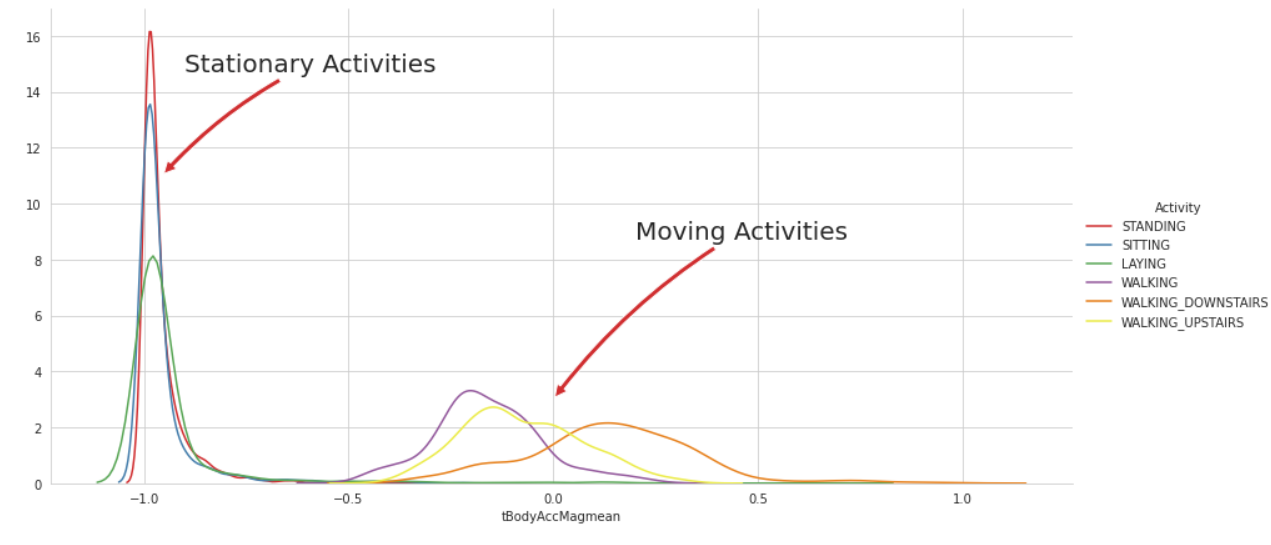


Figure 1.0: A visualization of the clustering in the initial data set [1]

Additionally, the notebook also provides visualization for a plot consisting of the density of that data on the y-axis against the variable 'tBodyAccMagmean' which clearly distinguishes the difference between kinetic and static activities performed in the data set.



Figure 1.1: A plot distinguishing stationary against moving activities (on the initial data set [1]) through a comparison of density on the y-axis and tBodyAccMagmean on the x-axis

2.1.3: Predictions of Data

# Chapter 3: Technical Overview

To tackle the classification problem at hand, it was decided to attempt to classify with four different supervised machine learning algorithms. The algorithms which were adopted are:

* Support Vector Machines
* K Nearest Neighbours
* Decision Trees
* Logistic Regression

The above choices all provide a different approach for classification and therefore, ensuring a very high probability rate for correct labelling.

## 3.1: Support Vector Machines

The objective of the support vector machine algorithm is to find a hyperplane in the Nth dimensional space which distinctly classifies the data points. Where the hyperplane can, for example: be a straight line in 2-dimensional space, and a plane in 3-dimensional space. Anything of greater dimensions would result in the hyperplane to be difficult to visualize.

To separate the two or more classes of data points, there are many possible hyper-planes that could be chosen. The objective of the support vector machine algorithm is to find a plane that has the maximum margin, that is, the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Why is a Support Vector Machine our best solution?

The data that has been collected consists of relationships of multiple dimensions. A support vector machine working with a radial kernel would provide the tools to determine the adequate margin considering a relationship of high dimensions.

### 3.1.1: Radial Kernel

## 3.2: K Nearest Neighbours

### 3.2.1: Ball Trees

The Ball Tree algorithm can be viewed as a 'Metric Tree' Algorithm. Metric Trees or Ball Trees organize and structure data points considering the dimensional plane in which the points are located. Therefore, using metrics, the points are not restricted to a finite dimensional relationship.

How does the Ball Tree work?

The data points are divided into two clusters in the shape of circles or hyper-spheres (spheres in essence), where each cluster is set to be a node of the resulting tree.

A point is selected. Considering this point, the algorithm searches for the farthest node to that point, where a vector which splits the data in half is plotted. All of the other data points are projected onto the vector via a simple transpose of their coordinates.

The median of the vector is found. With the median and a vector spanning from point 0 to N in mind, the algorithm plots two spheres. Where one sphere clusters from point 0 to the median M, and the other clusters from M to N. The process is repeated recursively considering the radius and median of the newly generated spheres.

Compared to KD Trees, the Ball Tree is not much of a victim to the Curse of Dimensionality. This is due to the lack of restriction to dimensions. This advantage compared to KD Trees, results in the Ball Tree algorithm for K Nearest Neighbours to be the most suitable for our data and implementation.

What is the Curse of Dimensionality?

The Curse of Dimensionality describes the problem in a high dimensional space where the data points when projected onto a vector or otherwise, are represented to be extremely close to one another and therefore the KNN algorithm finds it difficult to cluster the data accordingly.

### 3.2.2: KD Trees

## 3.3: Decision Trees

## 3.4: Logistic Regression

# Chapter 4: Code Overview

# Chapter 5: Testing and Results

# Chapter 6: Conclusion

# Chapter 7: References