Determining Individual Characteristics from Human Activity Data (iPhone Motion Sensor)

Introduction

Ofcom's figures indicate that in 2020 45 percent of households in the UK had two mobile phones. Published figures indicate there were over 53 million smartphones in the UK in May 2021 and over 2,700 million globally (O'Dea, 2021). It is therefore important to understand how smartphone data could be used, or abused. How could our privacy be impacted? The question I am interested in answering is - Can activity data recorded on a smartphone be used to infer individual characteristics about user, weight, height, gender, age, and if so, to what degree of accuracy?

Related Work

Research studies have established it is possible to recognise the type of activity that the user is undertaking to a high degree of accuracy. Researchers from University of California, Irving undertook experiments with a group of 30 volunteers within an age bracket of 19-48 years. The dataset is publicly available and has been used as a data source for many research papers (Reyes-Ortiz, 2012). Each person performed six activities wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, linear acceleration and angular velocity was captured at a constant rate of 50Hz.

On this similar dataset, support vector machine classifier and convolutional neural network machine learning models achieved almost 100 percent accuracy when determining activity type (Sun, 2019). Analysis established that within a short amount of time (1-1.5 min) the smartphone has enough data to determine what its user is doing (95%: 6 activities) or who the user is (Walking 94%: 30 participants) and even the basics of a person's specific walking style (Liftoff, 2019).

There is some existing analysis of this dataset published on GitHub and Kaggle websites. There is an initial exploratory analysis of the data subjects who participated in the trials (Shaar, 2019). There is also an example of predicting gender and average weight using this dataset using a single classification model (Huang, 2019). There is also published analysis of this time series data (Malekzadeh, 2019). This assessment builds on previous analysis, providing references where there is re-use of existing material.

Broader context

Cardiac abnormalities are one of the leading causes of deaths all over the globe. Smartphone-based systems may offer opportunities for real-time cardiac monitoring and early abnormality detection (Shabaan et. al, 2020). It is important to test the accuracy of predictive models utilising smartphone data.

A paper on wearable devices identified 423 unique devices from 132 different brands. With recent advances in mobile sensor technology, privately collected physical activity data can be used as an addition to existing methods for health data collection in research (Henriksen et. al, 2018). Given the popularity of wearable mobile sensors, it is important to determine whether smartphone activity data could provide a viable alternative.

If users can be personally identified by smartphone motion data the implications are that sensor data could be biographically significant legally, under the General Data Protection Regulations (Liftoff, 2019). Therefore data privacy concerns need to be addressed when data 'relates to or is 'obviously about' a particular individual. It is important that smartphone users understand the data that can be captured. This will allow informed choices to be made and privacy trade-offs to be transparent. Users may find monitoring useful for improving the quality health interventions.

Hypothesis

The hypothesis being assessed is whether personal characteristics of weight, height, age and gender can be predicted from motion sensor captured by a smartphone. This analysis evaluates a series of machine learning classifiers for prediction accuracy and time to compute, using a variety of subsets of the data based on activity type and applying feature engineering to the whole dataset.

Experiment

An experiment at Queen Mary University of London generated the dataset being reviewed. Time-series data was generated by accelerometer and gyroscope sensors. It was collected from an iPhone 6s kept in the participants' front pocket using SensingKit which collects information from Core Motion framework on iOS devices. All data collected in 50Hz sample rate. A total of 24 participants in a range of gender, age, weight, and height performed 6 activities in 15 trials, of varying length, in the same environment and conditions: downstairs, upstairs, walking, jogging, sitting, and standing. Full study conditions are provided by the researchers to allow repetition (Malekzadeh, 2019).

Modelling

Regression is an algorithm in supervised machine learning that can be trained to predict real number outputs. In this analysis I was interested in whether or not the individual could be identified as above or below average height, weight and age, not what the actual value was. Gender was only recorded as male or female, therefore I used classifiers. Classification is an algorithm in supervised machine learning that is trained to identify categories and predict in which category they fall for new values. The most common way to measure the accuracy of a classification model is by simply calculating the percentage of correct classifications the model makes on test data, after being trained on a training data set.

If there exists a hyperplane that perfectly separates the two classes, then the two classes linearly separable. Few examples of linear classifiers are Logistic Regression, Perceptron, Naive Bayes, Support Vector Machines, etc. Examples of non-linear classifiers are Decision Trees, K-Nearest Neighbour, Random Forest, non-linear-kernel Support Vector Machines, etc. K-Nearest Neighbour uses variable distance, Euclidean is most popular calculation used. This classifier is better for small data.

Logistic Regression is shallow learning (linear), Multi-Layer Perceptron is deep (non-linear). Deep learning refers to neural networks with multiple hidden layers that can learn increasingly abstract representations of the input data.

Multi Layer Perceptron (more than one layer) is suitable for non-linear separable data. However it is difficult to see relationship between input and output and may not scale well. A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a non-linear activation function. MLP utilises 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.

Summary of analysis

Following the CRISP-DM methodology, I researched the dataset to understand context and reviewed existing analysis. I performed exploratory data analysis, generating numerical and graphical summaries to understand the data associated to the trial participants. A Python code notebook providing access to regenerate the analysis is included as Appendix A.

There were 24 data subjects in the trial, 10 were females and 14 males. The raw data from the 15 activities was loaded and grouped into the 6 activities. Exploratory analysis was conducted to ascertain spread of height, weight, gender, age for the trial participants as a group, and then subdivided by gender. Graphs were generated for the whole group, then for males and females. Minimum, maximum and mean values were calculated for each attribute for the whole group, then for males and females.

I based my exploratory analysis of trail participants on the work performed by Shaar (Shaar, 2019). I reused and extended his work by cleansing the data to move code, which was a unique identifier and didn't add value to the analysis. I generated additional graphs splitting the data by gender, histogram/bar chart, correlation matrix and scatter/density plots. In addition, I produced a summary tables for minimum, maximum, average variable values for the whole dataset and also by gender.

For the trial output data, I added labels to allow opportunities to use supervised learning as well as unsupervised machine learning models and techniques. I cleansed, grouped, filtered, performed feature engineering and scaled the data in preparation. I performed mapping for trial participants to determine whether they were over average weight, height and age (setting the response variable to 1 for above, 0 for below). I ran a series of linear and non-linear, machine and deep learning models and evaluated them using a variety of metrics – accuracy, f1 score, time to compile. I produced a confusion matrix for the best performing model, showing error rate for true and false, positive and negative predictions.

The multivariate time-series has 12 features: attitude.roll, attitude.pitch, attitude.yaw, gravity.x, gravity.y, gravity.z, rotationRate.x, rotationRate.y, rotationRate.z, userAcceleration.x, userAcceleration.y, userAcceleration.z. The data was loaded with added identifiers for the experiment number, participant number and activity type. Exploratory analysis was conducted to ascertain class balance, hierarchical clustering and a sample of the timeseries data (userAcceleration.z) was examined by activity type. In total there were over 1.4m observations with 16 attributes. The datasets for the downstairs and jog were the smallest data sets, upstairs and standing next in order, with sitting and walking datasets the largest. A Python code notebook providing access to regenerate the analysis is included as Appendix B.

Best Subset Selection analysis was conducted on the whole dataset, using a Logistic Regression model. I repeated the Best Subset Selection analysis using two subsets of the data, on the Jogging activity type and on the Standing activity type. I evaluated the output using first RSS and R squared then Mallow's Cp (C_p), Akaike's Information Criteria (AIC) and Bayesian Information Criteria (BIC)and Adjusted R2. A Python code notebook providing access to regenerate the analysis is included as Appendix C.

I was able to run a limited selection of classification models on subsets of the data, split by activity type. I did scale the data, which improved the accuracy of the models. I did not apply feature engineering at this stage. I ran two Logistic Regression models, Decision tree and K-Nearest Neighbour against Jogging, Standing. Then I grouped activity types that indicated movement, 'Movement' is the analysis consisting of Upstairs, Downstairs, Walking trials and no movement named 'Static' in the analysis, consisting of the trial results for Standing and Sitting activity types. I used the models to predict the trial participant, and then if gender or above and below weight, age, height could be predicted. In addition to accuracy I tracked how long each model took to compute on Google Colab. Python code notebooks providing access to regenerate the analysis are included as Appendix D, E, F and G.

I extended the work produced by Y.C. Huang (Huang, 2019). I utilised his approach to feature construction and engineering based on that of ROYT's (T, 2019) allowing use of all complete dataset. In addition to gender and weight prediction I extended Huang's work to include mapping for height and age. I split the data into training and testing datasets, consistently for all eight classifiers and scaled the training data. To ensure the analysis was as comprehensive as feasible I chose to use a variety of linear and non-linear deep and machine learning models – Multi Layer Perceptron, Support Vector Machine Linear, Support Vector Machine - Radial Basis Function (RBF), Logistic Regression (L1), Logistic Regression (L2), Decision tree, Random Forest, K- Nearest Neighbour.

To undertake the assessment of whether individual characteristics could be ascertained by the models the response variable was adjusted. For all models accuracy and time to compute was assessed for determining trial participant, gender, weight, height and age. In addition to model accuracy scores, mean f1 score values were produced and a confusion matrix was generated for the model with highest accuracy, being Random Forest. A Python code notebook providing access to regenerate the analysis is included as Appendix H.

In addition, I also ran seven of the models against the time-series data with a sliding window algorithm. I performed channel-normalisation and scaling. I reused code developed by Mohammad Malekzadeh and Adam Martin (Malekzadeh, 2019) to load the data. I reused and adapted the sliding window and feature engineering code from Yu Guan (Guan, 2019). This was a limited assessment and didn't include variable prediction. A Python code notebook providing access to regenerate the analysis is included as Appendix I.

Results & Conclusion

Even when the limitations of the dataset are considered, it is reasonable to conclude that personal characteristics of weight, height, age and gender can be predicted to a high degree of accuracy when using machine learning models to evaluate human activity sensor data (circa 95%). Scaling the data improves model accuracy marginally but applying feature engineering significantly improves the computational speeds and accuracy rates, against the complete dataset.

The sample size of trial participants was small with less than 30. If the sample size is too small it may be difficult to detect what was intended (Kar & Ramalingam, 2013). The trial participants, and therefore the information gathered may not be reflective of the population, i.e. the speed of movements within younger individuals may not be reflective of an older population. Therefore any predictions on age, gender, weight or height, given the small sample may not be an accurate prediction of test error.

Trial output dataset - scaled data, no feature engineering

For trial data with the activity type of Standing, the Decision Tree Classification Model was marginally more accurate than K- Nearest Neighbour, averaging 99% rather than 97%. Models ranged in accuracy of predicting gender 73% to 99%, height 72% to 99%, age 70% to 99% and weight from 63% to 99% accuracy. K- Nearest Neighbour took significantly longer to compute using Google Colab resources, taking an average of 76 seconds across the five response variables, compared the models that all had an average of circa 9.5 seconds.

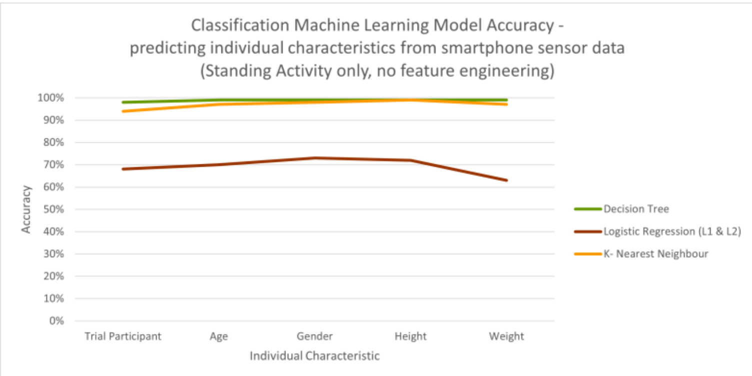


Figure 1 – Model Evaluation (Standing, no feature engineering)

For trial data with the activity type indicating no movement, i.e. standing or sitting, the Decision Tree Classification Model was marginally more accurate than K- Nearest Neighbour, averaging 99% rather than 98%. Models ranged in accuracy of predicting gender 63% to 100%, height 62% to 100%, age 66% to 99% and weight from 59% to 99% accuracy. Again, K- Nearest Neighbour took significantly longer to compute using Google Colab resources, taking an average of over 190 seconds across the five response variables, compared the models that all had an average of circa 19 seconds. The Static dataset (645k rows) was twice the size of the Standing dataset (306k rows). The processing speed reflected the increase, taking approximately 2 - 2.5 times longer.

Figures 2 & 3 – Model Evaluation (Static, no feature engineering)

For trial data with the activity type of Jogging, the K- Nearest Neighbour classification model had the highest accuracy rate. Models ranged in accuracy of predicting gender 62% to 97%, height 57% to 94%, age 60% to 95% and weight from 56% to 96% accuracy. Again, K- Nearest Neighbour took significantly longer to compute using Google Colab resources, taking an average of 22 seconds across the five response variables, compared the models that all had an average of 4 - 6 seconds. The Jogging dataset (134k rows) was the smallest dataset analysed without feature engineering.

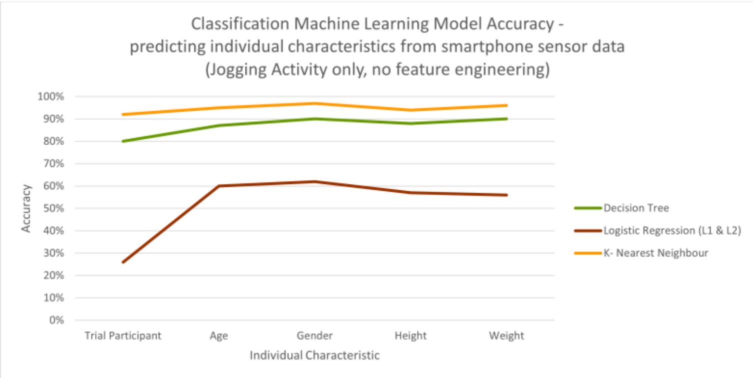
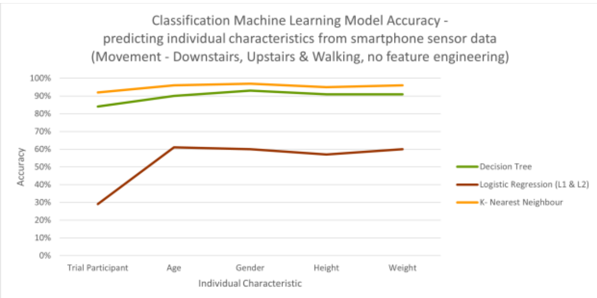
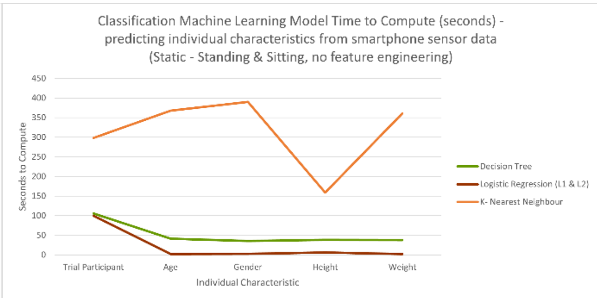


Figure 4 – Model Evaluation (Jogging, no feature engineering)

For trial data with the activity type indicating movement, i.e. upstairs, downstairs, walking. It did not include the Jogging activity type trial data. The K- Nearest Neighbour classification model had the highest accuracy rate. Models ranged in accuracy of predicting gender 60% to 97%, height 57% to 95%, age 61% to 96% and weight from 60% to 96% accuracy. Again, K- Nearest Neighbour took significantly longer to compute using Google Colab resources, taking an average of 315 seconds across the five response variables, compared the models that had an average computation time of 22 - 51 seconds. The Movement dataset (808k rows) was the largest dataset analysed without feature engineering.





Figures 5 & 6 – Model Evaluation (Movement, no feature engineering)

Feature engineering on scaled data

Where feature engineering was applied, performance of the models varied significantly with the Random Forest Classifier having the highest accuracy for each variable. Models ranged in accuracy of predicting gender 70% to 97%, height 62% to 97%, age 63% to 94% and weight from 61% to 96% accuracy.

The Random Forest classification model consistently performed with highest accuracy for predicting trial participant, age, gender, height and weight. However computational speed of the Random Forest classification model was significantly slower, averaging 37 seconds versus the other models that all computed with an average of less than 1 second. The Support Vector Machine Linear model consistently performed with lowest accuracy, indicating that the data was non-linear.

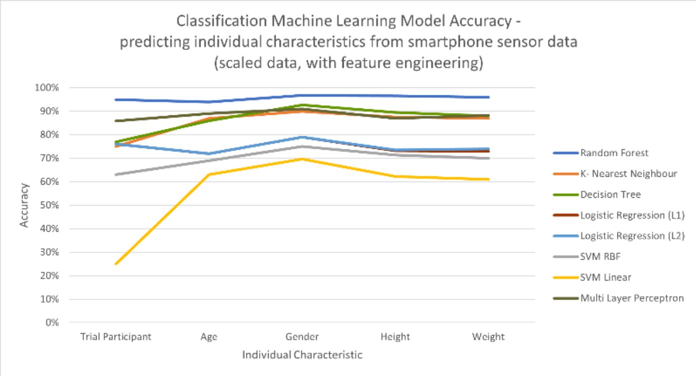


Figure 7 – Model Evaluation (Accuracy)

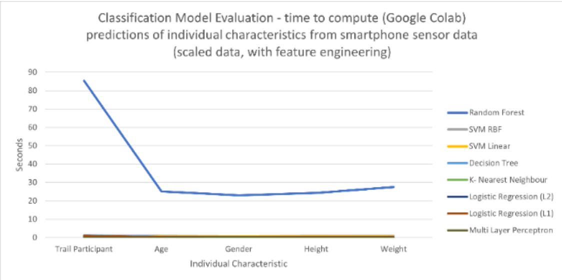


Figure 8 – Model Evaluation (Speed – All Models)

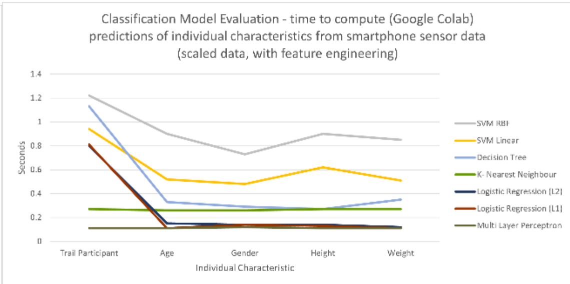


Figure 9 – Model Evaluation (Speed – Fast Models)

Time series, feature engineering, sliding window

On the time series, a sliding window algorithm was reused (Guan, 2019). With channel normalisation but even without feature engineering the models compute much faster. The Random Forest model had the highest accuracy at 95% and also took the longest time to compute, but just 0.29 seconds. The lowest accuracy was 70% from K- Nearest Neighbour with a total computational time of 0.01 seconds. When the feature engineering was applied all models produced an accuracy rate of 100%. However, the model was using a time series, sliding window to predict the next frame rather than being trained and tested on a response variable so cannot be compared with the other models and is not an indicator of whether individual characteristics could be determined accurately.

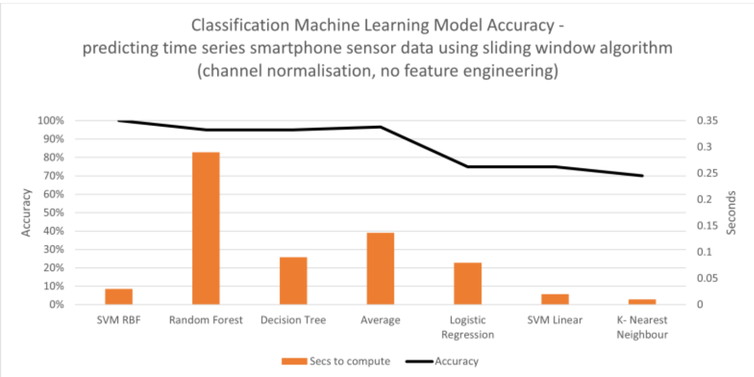


Figure 10 – Model Evaluation (Sliding Window Algorithm)

Evaluate success
The high average accuracy rate of the Decision Tree model, at 99%, which was trained and tested without feature engineering is worthy of note. Particularly as it was trained on activity types indicating no movement, like sitting and standing. Even when not moving, the smartphone sensor data can be used to accurately determine if the trial participant was above or below average weight, height and age. The K-Nearest Neighbour classification model had the highest accuracy rate on activity types that indicated movement, jogging, walking, going upstairs or downstairs, with an average accuracy of 95%.

As the Decision Tree favours non-linear separable data and the K-Nearest Neighbour favours a data structure that is linearly separable, the results may indicate the non-movement trial data is more non-linear than the movement data is linear. The models took significantly longer to compute than the models that utilised feature engineering, ranging from an average of 10 seconds to 315 seconds.

Once feature engineering had an impact on model accuracy, with rank order changing depending on the encoding technique applied to the raw data. The model accuracy using the complete dataset with feature engineering, wasn't as high as the analysis where activity types were split out without feature engineering. This possibly indicates the difference of data structure between the trials. Where feature engineering was applied the Random Forest model performed with highest accuracy. The Random Forest likely performed better than the Decision Tree model due to the dimensionality of the data being high and decision boundary being smooth. Decision tree models can easily overfit, cope poorly with high-dimensionality data and have a block effect to decision boundaries.

Even with feature engineering applied, the Random Forest took the longest time to compute of all the models. Again, the speed of the Support Vector Machine - Radial Basis Function (RBF) model indicates that the complete dataset with feature engineering may have a non-linear structure. There does not seem to be a relationship between model accuracy and computation time, instead the relationship is between data structure and model (linear or non-linear).

Random Tree confusion matrices for gender, weight and height had a higher number of false positives and age a higher number of false negatives. The false positives incorrectly predicted that gender was male or weight and height were above average. The false negative incorrectly predicted an age lower than the average. Full analysis is included in Appendix H. These results reflect the dataset bias as there were more males that females in the trail, and the men were the oldest, heaviest and tallest participants.

Future implications
Due to the similarities of the data structure within activity types, the data could have been subdivided prior to feature engineering. It is expected that model accuracy would have been improved further. The number of participants and the diversity of weight, height, age and gender could have been increased. This would improve the quality of the analysis, modelling and any conclusions.

Feature engineering development requires domain knowledge of the data, which I didn't have. More work could be undertaken in this area. The models I produced were based on the feature engineering from Y.C. Huang's code (Huang, 2019), who reused a calculation produced by ROYT (T, 2019). I also reused the human activity recognition sliding window algorithm from Yu Guan (Guan, 2019).

Google Colab resources limited the analysis feasible within a reasonable timescale. Feature engineering was required to allow the models to run efficiently on the complete dataset. Additional computation resources would be needed to analyse the complete dataset without dimension reduction. In a real-world scenario it may be necessary to compromise model accuracy in favour of computational speed.

Feature importance analysis wasn't complete enough to include within the summary findings report but is included in the Appendices for completeness. The calculated mean f1 score did not vary significantly with the exception of Support Vector Machines prediction for weight on the dataset where I performed feature engineering. Additional analysis of the confusion matrices could be undertaken.

Reflections

The process, technologies and methodologies I used involved a mixture of new and previously used. All of the approaches were still new to me. Being honest this assignment, like all the others, has been challenging yet rewarding. Even reminding myself how to run the GitLog file took time and persistence. Figuring 'stuff' out on my own and being resilient feels uncomfortable and difficult. In terms of personal and professional development I really enjoy, and hate, the learning process. I struggle being out of my comfort zone, feeling very anxious, stressed and frustrated at times. I need time to absorb information and be able to gain confidence in the topics being taught. Therefore managing my time well has been very important.

Developing trusted relationships with other students has been very useful particularly in determining if I've understood the material in the way that was intended, or at least they understood it. Appreciating that others have a different learning approach has also been valuable and quite insightful. Our differences as humans make us unique and recognising there are more than several 'right' ways has been eye-opening. In future I'd like to think I will be more flexible with how I define tasks for my team to allow them more scope for personal style. When planning learning and development tasks in future I will make sure to allow sufficient time for information to be assimilated, whether for myself or others.

Following feedback from previous assignments, I took more time to follow the Scientific Method and used that format for this report. I attempted to be clearer on how I had identified the problem, approached it, defined it, then selected to model it, evaluate how successful this had been and conclude what future work could be undertaken. I took more time, and words, to explain the rationale for my decisions.

I used the CRISP-DM best practice methodology. The lifecycle contains six phases; Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, Deployment. The process allows for iteration between stages and I did move between data preparation, modelling and evaluation extensively. This lead to a bit of yo-yo-ing. The analysis could have absorbed as much time as I was prepared to give. During the development of my report I cut back the content significantly to ensure it was focused.

I created a GitHub repository using the cookiecutter Data Science template in GitHub. I used Git for version control. I developed models using Python on Google Colab with data stored on Google Drive. The cookiecutter folder structure was useful for organisation and keeping order of my work. Google Colab provided easy access to sklearn library, keras and tensorflow code. I was not able to set up RStudio locally with the Python extensions and 'reiterate' library using a virtual environment.

I enjoyed learning Python and the Save As and Find and Replace features within Google Colab were extremely useful. Google Colab resources had limitations and signing on to the Google Drive for each notebook felt unnecessary. Without feature engineering I was unable to run all the models using Google Colab resources even when the activity types were split to reduce the size of the dataset.

I needed to split the code into separate notebooks to keep the size of the Python notebooks manageable. Uploading files directly from Google Colab needed the GitHub repository to be set to 'public'. Change of access and visibility needs password approval and additional text so switching this on and off was fiddly, although not difficult. As a result I limited my commits to when there was a reasonable amount of work to update within the repository. I would recommend Google Colab based on my experience. Although I had not coded before in Python, I did find it reasonably intuitive but handling code errors in Jupyter notebooks was not as easily identifiable or rectifiable as R. Some tasks were beyond my skillset and I had to accept defeat. There were some sections of code I was unable to generate due to library problems in Google Colab (dtiadistance gave a C language error, for example).

The two Logistic Regression Models produced very similar results and I possibly could have removed one had I realised that early enough. It was only in the final review I realised how similar they were.

It is important that smartphone users are aware how motion sensor data captured may be used by modelling techniques to reveal individual characteristics.

Detailed findings

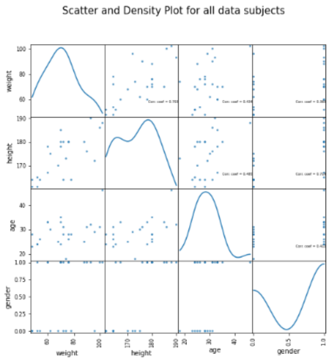
There were 24 data subjects in the trial, with average weight of 72kg (11 stones 4lb), average age of 29 and average height of 174cm (5ft 8.5') Fourteen males and ten females participated.

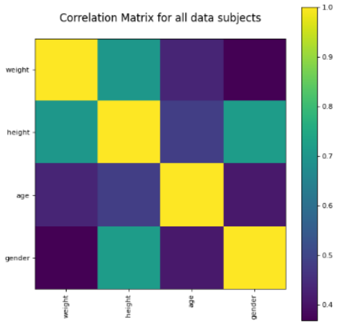
The histograms for the data show the variation in weight, height, age and gender of the participants in the trail. The average weight of all trail participants was 72.125kg, average height was 174.2cm and average age of 28.79 years. Fourteen males and ten females participated.

Average age of the female participants was 26.2 years, average height was 166.8cm and average weight was 65.2kg. Average age of the male participants was 30.6 years, average height was 179.5cm and average weight was 77.0kg.

The oldest, heaviest and tallest participants in the trial are males. The lightest, smallest and youngest are all female participants. There is a correlation between height and weight and a weaker correlation between age and weight for female participants. For male participants, the only and strongest correlation is between height and weight.

There are additional charts for males and females. A Python code notebook providing access to regenerate the analysis is included as Appendix A.





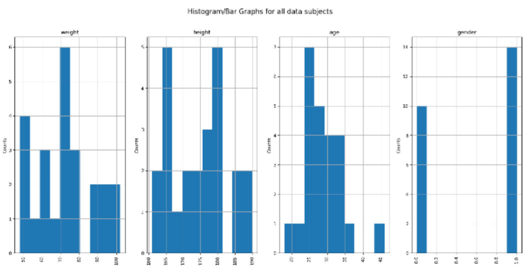


Figure 11 – Scatter Plot (All Trial Participants)

Figure 12 – Correlation Matrix (All Trial Participants)

Figure 13 – Histogram (All Trial Participants)

Best Subset Selection analysis was conducted on the whole dataset to determine the variables required to accurately predict activity type, i.e. which variables accounted for the highest variance. A further two subsets of the data where activity type was known, Jogging and Standing were analysed to determine the variable required to accurately predict trial participant. All the Best Subset Analysis was undertaken without scaling or feature engineering, using a Logistic Regression model. Due to the high number of recommended predicter variables there was little opportunity to reduce the dataset.

There are additional Best Subset Analysis charts for Jogging and Standing. A Python code notebook providing access to regenerate the analysis is included as Appendix B.

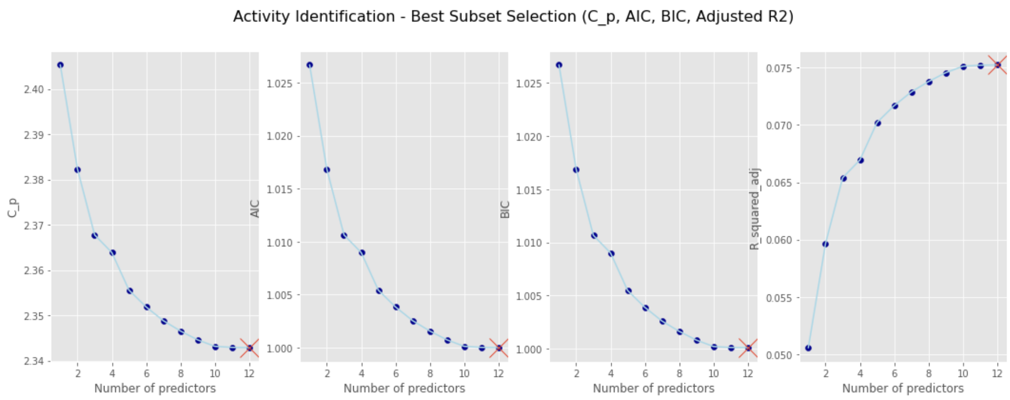


Figure 14 – Best Subset Selection Evaluation (All Data)

Python code notebooks providing access to regenerate the analysis are included as Appendix D, E, F and G. Line graphs included within this report and the tables below are included as Appendix J - Charts Machine Learning.

Model Evaluation - Standing	Trial Participant			Age			Gender			Height			Weight			Average		
	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute
Decision Tree	98%	98%	19.1	99%	99%	7.75	99%	99%	7.1	99%	99%	6.7	99%	99%	7.06	99%	99%	9.54
Logistic Regression (L1)	68%	67%	41	70%	69%	0.84	73%	72%	1.08	72%	72%	3.78	63%	61%	1.22	69%	68%	9.58
Logistic Regression (L2)	68%	67%	41	70%	69%	0.87	73%	72%	1.07	73%	72%	3.94	63%	61%	1.19	69%	68%	9.61
K- Nearest Neighbour	94%	94%	93.96	97%	97%	94.09	98%	98%	89.12	99%	99%	14.95	97%	97%	89.41	97%	97%	76.31
Average	82%	82%	48.77	84%	84%	25.89	86%	85%	24.59	86%	86%	7.34	81%	80%	24.72	84%	83%	26.26

Table 1 – Model Evaluation (Standing, no feature engineering)

Model - Static (Sitting & Standing) Evaluation	Trial Participant			Age			Gender			Height			Weight			Average		
	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute
Decision Tree	99%	99%	33.78	99%	99%	15.94	100%	100%	13.41	100%	100%	14.45	99%	99%	15.8	99%	99%	18.68
Logistic Regression (L1)	56%	53%	84.57	66%	65%	1.87	63%	61%	1.94	62%	61%	8.5	59%	54%	1.85	61%	59%	19.75
Logistic Regression (L2)	56%	53%	82.83	66%	65%	1.88	63%	61%	1.87	62%	61%	8.52	59%	54%	1.8	61%	59%	19.38
K- Nearest Neighbour	95%	95%	200.67	97%	97%	242.39	98%	98%	231.73	100%	100%	52.29	98%	98%	229.91	98%	98%	191.40
Average	77%	75%	100.46	82%	82%	65.52	81%	80%	62.24	81%	81%	20.94	79%	76%	62.34	80%	79%	62.30

Table 2 – Model Evaluation (Static, no feature engineering)

Model Evaluation - Jogging	Trial Participant			Age			Gender			Height			Weight			Average		
	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute
Decision Tree	80%	79%	14.85	87%	87%	4.88	90%	90%	4.86	88%	88%	5.12	90%	89%	4.59	87%	87%	6.86
Logistic Regression (L1)	26%	22%	20.95	60%	55%	0.38	62%	56%	0.38	57%	55%	1.23	56%	45%	0.42	52%	47%	4.67
Logistic Regression (L2)	26%	22%	20.73	60%	55%	0.39	62%	56%	0.38	57%	55%	1.22	56%	45%	0.38	52%	47%	4.62
K- Nearest Neighbour	92%	91%	23.35	95%	95%	24.99	97%	97%	24.04	94%	94%	16.54	96%	96%	24.24	95%	95%	22.63
Average	56%	54%	19.97	76%	73%	7.66	78%	75%	7.42	74%	73%	6.03	75%	69%	7.41	72%	69%	9.70

Table 3 – Model Evaluation (Jogging, no feature engineering)

Model Evaluation - Move (Upstairs, Downstairs, Walking)	Trial Participant			Age			Gender			Height			Weight			Average		
	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute
Decision Tree	84%	84%	105.85	90%	90%	41.08	93%	92%	35.27	91%	91%	38.38	91%	91%	37.74	90%	90%	51.66
Logistic Regression (L1)	29%	24%	98.54	61%	57%	2.44	60%	55%	2.59	57%	56%	6.33	60%	51%	2.4	53%	49%	22.46
Logistic Regression (L2)	29%	24%	103.75	61%	57%	2.43	60%	59%	2.58	57%	56%	6.4	60%	51%	2.39	53%	49%	23.51
K- Nearest Neighbour	92%	92%	297.98	96%	96%	367.95	97%	97%	390.19	95%	95%	158.77	96%	96%	360.62	95%	95%	315.10
Average	59%	56%	151.53	77%	75%	103.48	78%	76%	107.66	75%	75%	52.47	77%	72%	100.79	73%	71%	103.18

Table 4 – Model Evaluation (Movement, no feature engineering)

A Python code notebook providing access to regenerate the analysis is included as Appendix H.

Model Evaluation (with Feature Engineering)	Trial Participant			Age			Gender			Height			Weight			Average		
	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute	Accuracy	mean f1 score	Secs to compute
Random Forest	95%	95%	85.39	94%	94%	25.13	97%	97%	23.02	97%	97%	24.26	96%	96%	27.55	96%	96%	37.07
Multi Layer Perceptron	86%		0.11	89%		0.11	91%		0.12	87%		0.11	88%		0.11	88%		0.11
Decision Tree	77%	77%	1.13	86%	86%	0.33	93%	93%	0.29	89%	89%	0.27	88%	88%	0.35	87%	87%	0.47
Logistic Regression (L1)	76%	76%	0.8	72%	71%	0.11	79%	79%	0.14	73%	73%	0.13	73%	71%	0.11	75%	74%	0.26
Logistic Regression (L2)	76%	76%	0.81	72%	71%	0.15	79%	79%	0.14	74%	73%	0.14	74%	71%	0.12	75%	74%	0.27
K- Nearest Neighbour	75%	75%	0.27	87%	87%	0.26	90%	90%	0.26	88%	88%	0.27	87%	87%	0.27	85%	85%	0.27
SVM RBF	63%	63%	1.22	69%	68%	0.9	75%	74%	0.73	71%	71%	0.9	70%	66%	0.85	70%	68%	0.92
SVM Linear	25%	21%	0.94	63%	57%	0.52	70%	70%	0.48	62%	62%	0.62	61%	47%	0.51	56%	51%	0.61
Average	72%	69%	11.33	79%	76%	3.44	84%	83%	3.15	80%	79%	3.34	80%	75%	3.73	79%	77%	5.00

Table 5 – Model Evaluation (All data, feature engineering)

A Python code notebook providing access to regenerate the analysis is included as Appendix I.

Model Evaluation - Sliding Window	Accuracy	mean f1 score	Secs to compute
SVM RBF	100%	100%	0.03
Random Forest	95%	95%	0.29
Decision Tree	95%	95%	0.09
Average	97%	97%	0.14
Logistic Regression	75%	67%	0.08
SVM Linear	75%	67%	0.02
K- Nearest Neighbour	70%	57%	0.01

Table 6 – Model Evaluation (Sliding Window, no feature engineering)

References:

Y. Guan, 2019. *The role of feature Engineering on HAR* (GitHub) [ONLINE] Available at: https://github.com/yuguan1/CSC8111_code/blob/master/ML_day3/ML4HAR_1.ipynb [Accessed 14 January 2022].

A. Henriksen, M.H. Mikalsen, A.Z. Woldaregay, M. Muzny, G. Hartvigsen, L.A. Hopstock, S. Grimsgaard, 2018. *Using Fitness Trackers and Smartwatches to Measure Physical Activity in Research: Analysis of Consumer Wrist-Worn Wearables (Journal of Medical Internet Research)* [ONLINE] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5887043/ [Accessed 16 December 2021].

Y.C. Huang, 2019. *My analysis on motion sensor data (Kaggle)* [ONLINE] Available at: https://www.kaggle.com/teaprint/my-analysis-on-motion-sensor-data [Accessed 16 December 2021].

D. Liftoff, 2019. *What Does Your Smartphone Know About You? (Kaggle)* [ONLINE] Available at: https://www.kaggle.com/morrisb/what-does-your-smartphone-know-about-you [Accessed 16 December 2021].

Kar, S.S. & Ramalingam, A., 2013. Is 30 the magic number? Issues in sample size estimation. *National Journal of Community Medicine,* 4(1), p. 175.

M. Malekzadeh, 2019. *MotionSense Dataset: Smartphone Sensor Data (Kaggle)* [ONLINE] Available at: https://www.kaggle.com/malekzadeh/motionsense-dataset [Accessed 16 December 2021].

S. O'Dea, 2021. *Number of mobile phones per household in the United Kingdom (UK) in 2020 (Statista.com)* [ONLINE] Available at: https://www.statista.com/statistics/387184/number-of-mobile-phones-per-household-in-the-uk/ [Accessed 16 December 2021].

J.L. Reyes-Ortiz, D. Anguita, A. Ghio, L.Oneto, X. Parra, 2012*. Human Activity Recognition Using Smartphones Data Set (UCI Machine Learning Repository)* [ONLINE] Available at: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones [Accessed 16 December 2021].

R. T, 2019. *A simple Features DNN using TensorFlow (Kaggle)* [ONLINE] Available at: https://www.kaggle.com/talmanr/a-simple-features-dnn-using-tensorflow [Accessed 12 January 2022].

S.T.A. Shaar, 2019. *Starter: MotionSense Dataset : 8c09b08d-5 (Kaggle*) [ONLINE] Available at: https://www.kaggle.com/salahuddinemr/starter-motionsense-dataset-8c09b08d-5 [Accessed 16 December 2021].

M. Shabaan, K. Arshad, M. Yaqub, F. Jinchao, M.S. Zia, G.R. Boja, M. Iftikhar, U. Ghani, L.S. Ambati, R.Munir, 2020*. Survey: smartphone-based assessment of cardiovascular diseases using ECG and PPG analysis (BMC Medical Informatics and Decision Making)* [ONLINE] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7392662/ [Accessed 16 December 2021].

X. Sun, 2019. *Human Activity Recognition Using Smartphones Sensor Data (medium.com)* [ONLINE] Available at: https://medium.com/@xiaoshansun/human-activity-recognition-using-smartphones-sensor-data-fd1af142cc81 [Accessed 16 December 2021].

G.Wang, Q. Li, L. Wang, W. Wang, M. Wu, T. Liu, 2018. *Impact of Sliding Window Length in Indoor Human Motion Modes and Pose Pattern Recognition Based on Smartphone Sensors (Sensors – Basel, Switzerland)* [ONLINE] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6021910/ [Accessed 16 December 2021].