June 7, 2019

I certify that this is all my own original work. If I took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in my submission. I will show I agree to this honor code by typing "Yes": *Yes*.

THE ANALYSIS OF SINGLE CHEST-MOUNTED ACCELERATER DATA SET

BY CLASSIFICATION ALGORITHM

Student ID: s3677615

Student Name: Shuo Wang

Affiliations: RMIT

Contact Details: s3677615@student.rmit.edu.au

Table of Contents

Abstract:

This report aimed to classify seven different types of human activities, such as walking, working at a computer, into target labels by using the data collected from a wearable accelerometer mounted, x, y and z-axis elements with classification algorithm, K Nearest Neighbor and Decision Tree. The dataset was obtained from UCI machine learning repository and processed by data retrieval and preparation, data exploration and data modelling. Moreover, hill climbing with the score of K Nearest Neighbor was used to make feature selection, and K-Fold cross-validation and nested loop have been used in tuning parameters for modifiers. Overall the accuracy of the prediction results was about 75 percent for both K Nearest Neighbor and Decision Tree algorithms. The challenges include unbalanced classes and a load of data on a computer's performance.

1 Introduction

Activity recognition is an emerging field that developed in health care and the human-machine interaction domain with ubiquitous computing, context-aware computing, and multimedia [1]. Accelerometer sensor has been widely used to collect data generated from users' behaviour because it has many benefits, such as low energy consumption, low cost, and fewer weights in real-time. A Bluetooth wearable accelerometer sensor that is worn in the breast of people can gather the data from the direction of movement by the X-axis, Y-axis and Z-axis. The data are collected from fifteen participants to perform seven activities, including working at the computer, standing up, walking and going up/down stairs, standing, walking, going up/down stairs, walking and talking with someone, and talking while standing. For each participant, a person was asked to annotate the sequential order of the activities, and the data is collected sequentially with 52 Hz sampling frequency of the accelerometer.

This relevant dataset was posted in the UCI machine learning repository in 2014, and the dataset is Activity Recognition from Single Chest-Mounted Accelerometer [2]. Fifteen CSV format files were collected from 15 participants with the real data type and without missing values, and each file contains a sequential number, x acceleration, y acceleration, z acceleration, and labels. The direction is shown in Figure 1. The primary purpose of this report is to predict people's motion patterns from the data collected from x acceleration, y acceleration, and z acceleration with classification algorithms.

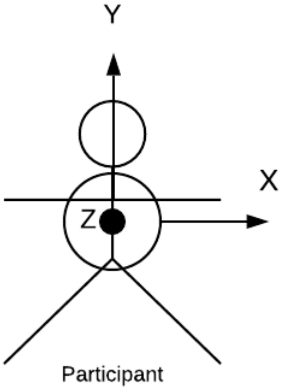


Figure 1. Coordinate System

2 Findings

Task 2.1: Retrieving and Preparing the Data

The purpose of this part is data preprocessing that is a process of collecting and cleaning, including missing values, typos, extra whitespaces, Case sensitive and sanity checks for impossible values. Firstly, all the fifteen files were loaded respectively and be appended in a list, and then it was merged into one Pandas framework by concatenation function. Besides, whether the dataset was imported ways correctly with matching the original dataset and data type and any mistake was checked. Finally, according to different situations, a variety of processing methods will be implemented, such as deleting the whole column or filled in some meaningful value.

Approximately 4000 wrong label data is 0, which should be deleted or filled in other meaningful values because the labels should be between 1 to 7. Because it is a sequential data that executed a series of actions[2], the wrong values can be seen similar to the neighbours' labels, below or above this label. Thus, for each label of 0, the neighbours will be checked, and there are a few situations:

1. The closest two values(below and above) for one active 0 label are the same. This 0 will be changed as same as neighbours' value.
2. If the index of one neighbour side is out of the index boundary, this 0 label will be filled to another neighbour's side.
3. The closest two values are different, which probably happened at the interaction between two labelled boundaries. The 'below' neighbours will be selected all the time in this situation.

Task 2.2: Data Exploration
This step will process the dataset by generating different appropriate graphs that explore each data feature and find a relationship with different attributes.

Task 2.2.1: Single Data Feature
First of all, from the histogram of Figure 2 (bins size: 240), it is clear to see that x acceleration and z acceleration is mainly located on both sides of 2000, and x acceleration has a wide range of variability. Moreover, y acceleration is primarily located in 2400, which has a higher density.

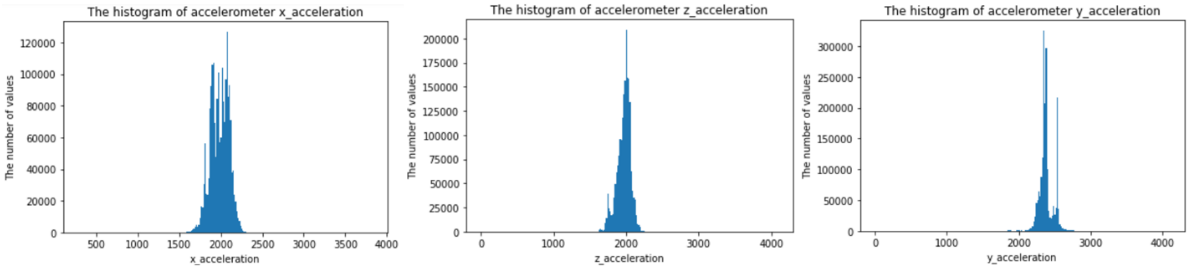


Figure 2. x, y, z-axis Exploration

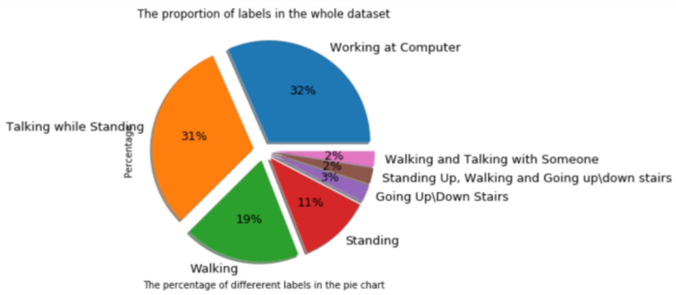


Figure 3. The proportion of labels

From Figure 3, it gives information about in these 15 datasets, the label of working at the computer and talking while standing provided a more significant part of proportion, which is 32% and 31%, respectively. For walking and talking with someone, standing up, walking, going up/down stairs and going up\down stairs have been provided with a tiny part of the proportion that is only provided 2%, 2% and 3%, separately. This unbalanced labelled data(class imbalance or class skew) could dramatically decrease the prediction performance in the data modelling part.

Task 2.2.2: Relationship Between Pairs of Attributes
As Figure 1 shows, Z-axis represents the axis concordant to the direction of movement and the plane defined by the X and Y axis lies on the body of the person [1].

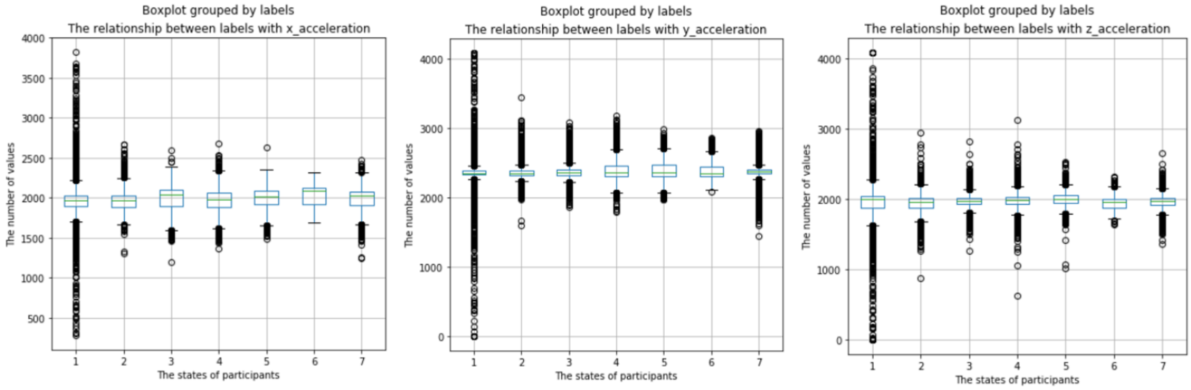


Figure 3. The Relationship Labels with x, y, z-axis

From Figure 3, it is clear to see that when people were working at the computer (label 1), more points vibrate from 1 to 4000 proximately in the x, y, and z-axis, compared with other activities. Pierluigi et al. claim that for the data collection, people were asked to work at a computer during the 86 minutes [1]. Thus, participants needed to adjust the position to relieve tense muscles in different paths to make them comfortable. A survey also proposed that an ergonomic chair that is suitable people gesture will alleviate the pain during working at the desk [3]. Moreover, in the y acceleration graph, this graph gives information about when people walking, going up and down stairs and walking and talking with someone (labels 4, 5, 6), many value points have a higher vibration compared with other labels. Firas et al. claim that because human walking has a peculiar straight-legged style, the body's centre of mass moves up and down with each step [4]. Besides, it is clear to see that talking while standing (label 7) has a broader range of data points distribution and a closer gap between the 25th percentile and 75th percentile. Because when people are talking with others, people are not only just talking with someone but also body language, such as movement and hand gestures [5]. Finally, from the x accelerate figure, we can see that when people were standing (label 3), the accelerometer is always shifted to one side, which is most of the circle points located in the value of about 1500, but the mean value to the other side. Some studies suggest that standing with both legs makes us more alert while standing with one leg mean we are rather calm and off guard, and usually, we can also see whom someone attracted to by their legs. When we stand with one leg, the other leg that has less weight tends to point to the people that we are attracted to [7].

Except above plausible hypothesis, some interesting relationships can also be visualized. For example, in the x acceleration diagram, when people were talking and walking with someone, they can keep bodies relatively stable on the x-axis level. Because this accelerometer is uncalibrated, in the y-axis, probably some errors lead to all the mean value is over than 2000, which is dramatically different between other relationship diagrams.

3 Methodology

The dataset that describes seven types of activities as labels with 3 data features was downloaded from the UCI Machine Learning Repository. These three data features are numerical values. Because all the data has a similar numeric scale, there is no need standardization to formate data features x, y, z-axis. In order to predict the labels, the K nearest neighbour and decision tree models have been used, and the type of activities will be selected as a target value. After the target feature was recorded by another variable, the target value was deleted from the raw data to prevent data leakage during model training.

3.1 Feature Selection by Hill climbing

Feature selection is also called variable selection or attribute selection that is the process of selecting a subset of relevant features. It can not only improve the performance of the modifier but also provide faster and more cost-effective predictors that reduce the burden of computing resource and a better understanding of the model based on processed data. Generally, there are three types of feature selection algorithms, including filter methods, wrapper methods and embedded methods [7]. This report mainly implemented a wrapper method that considers the selection of a set of features as a search problem. Different combinations of data feature are prepared, evaluated and compared to other combinations, and a predictive model will be used to evaluate a combination of features according to the score of model accuracy. A random hill-climbing algorithm is a kind of wrapper method.

Hill climbing examins attributes one by one forward randomly and selects the best successor node under the evaluation function, which can enhance the correct rate. Firstly, the order of input data features will be generated by random seeds. And then, appended data features will be input into an algorithm model (KNN) to check whether there is an improvement. If yes, this data feature will be recorded and be seen as an algorithm with positive contributions. Otherwise, this data feature will be dropped out.

One of the drawbacks of this algorithm is the only locally optimal solution because it depends on the random seed. The optimum accuracy rate is calculated by n*(n-1)/2, which will greatly increase the computational burden in a large number of data features. However, the data set of accelerometer only has 3 data features. It does not need to worry about this. Moreover, the decision tree will make feature selection automatically based on information gain or Gini index when the algorithm split the data in the split node.

3.2 The K Nearest Neighbor with K-Fold Cross-Validation
After feature selection, K-Fold cross-validation implemented to tune the parameters of K Nearest Neighbor (KNN) and test the performance of the modifier on average. Because the selection of models and parameters will largely depend on how to divide the training set and the test set, which means under different partitioning methods, the corresponding optimal degree is also different, a good division between training and test sets is important. Moreover, it will use all the data available in the dataset.

K-Fold cross-validation can split data into the K folders, and each time randomly, it will use k-1 values for training a model and 1 part will be used to test results. Thus, K-Fold cross-validation will train and test the dataset K times. The value of K depends on how many datasets will be tested. Because the dataset of activity recognition from single chest-mounted accelerometer has a huge number of values, three folders chose in here. After that, it will be fit a model on the training set and evaluate it on the test set. Then, it will retain the evaluation score and discard this model. Finally, with the K times iterations, a higher average score will be selected with relevant parameters.

For the KNN classifier, eight parameters can be tuned. However, in this report, only the number of neighbours, weights and power were considered. N_neighbors relates to the number of neighbours that have been tested to judge the classification waiting for a predicted value based on some kinds of distance calculation algorithms. The weight includes 'uniform and distance'. Uniform means all the neighbours for target value is weighted equally. However, the distance means the algorithm will consider the contribution by the value of the distance. The closer value will have a higher contribution. The power parameter can implement different distance calculation algorithms. For example, if p equal 1, it will use Manhattan distance. If p equals 2, it will take user Euclidean distance, and if p is greater than 2, Chebychev distance will be considered [8].

$$distance = (\sum_{k=0}^{n} |x_i - y_i|^{1/p})$$

Where $X = (x_1, x_2, x_3 \dots x_n)$ and $Y = (y_1, y_2, y_3 \dots y_n)$ for each dimensionality in the dataset.

The parameters were tuned by four nested for loop to greedy search each possible better combinations—three loops for the parameters in KNN classifier and one loop for K-Fold cross-validation. Because the dataset of the accelerometer is huge, parameter adjustment adopts unrefined adjustment first, then fine adjustment. A higher range of parameter was selected first. After we know roughly the exact parameters interval, a narrow parameter range with iteration in that higher score interval was implemented to reduce the running load of the computer. For instance, it tuned n_neighbors parameter from 1 to 100 as incremental value five each iteration. After we know the approximate interval of high accuracy, for example, incremental value 2 for each iteration will be selected.

3.3 The Decision Tree with K-Fold Cross-Validation

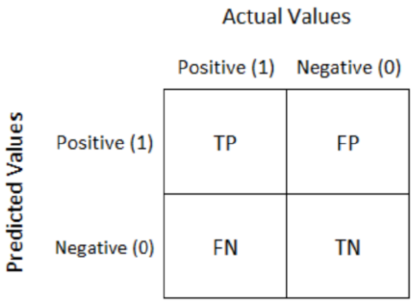
A decision tree is drawn upside down with its root at the top, which can respectively divide datasets into many parts by choosing different conditions in each step to separate data samples in each part. The primary purpose is to make each leaf node as pure as possible. Because the characteristics of the decision tree algorithm, it can arrive at a higher accuracy rate in training level, however, lower score in the test data, which is called overfitting. Pruning is one way to decrease overfitting by limiting the growth of branches that removes the branches that make use of features having low importance. It will reduce the complexity of the tree and increase its predictive power to unseen data (test data).

There are over ten affecting factors in the modifier, but in this report, only four parameters were considered, including criterion:[‘gini', 'entropy'], max_depth, min_samples_split, min_samples_leaf. For criterion, the function to measure the quality of a split has been tested by 'Gini' an 'entropy' that can consider the degree of purity for each data feature. Min_samples_leaf means the minimum number of samples that were needed at a leaf node. Max_depth means the maximum depth of the decision tree and min_samples_split means the minimum number of samples required to split an internal node.

For tuning parameters, it used four nested for loop iteration for each parameter with one loop K-Fold cross-validation. However, because the dataset has a considerable amount of data, it will take a long period to train and test dataset. Thus, it used a similar way to KNN. Tuning the parameters with a large interval value for the first time to guide a direction, and then tunning the parameters in more precision ways. Because it limited by processor performance, it may only obtain an optimal local performance with relevant parameters.

3.4 Evaluation metrics

The designed algorithm will only consider the highest score with relevant parameters. These relevant parameters will be selected to generate confusion matrix and classification report to check the modifier performance. In the confusion matrix, it will put predicted values and actual values in two ways to create a two-dimensionality matrix to calculate results. From Figure 4, it is clear to see that accurate value is shown in true positive (TP) and true negative (TN), which means the data point is predicted to true, and the actual value is true, and the model predict the value as false, and the actual is false as well.



	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Figure 4. Confusion Matrix

Vice versa. False negative (FN) and false positive (FP) mean the wrong prediction with the test dataset.

Recall: for all the positive classes, how many datasets were predicted correctly.

$$Recall = \frac{TP}{TP + FN}$$

Precision: in the predicted values in positive class, how many datasets were predicted correctly.

$$Precision = \frac{TP}{TP + FP}$$

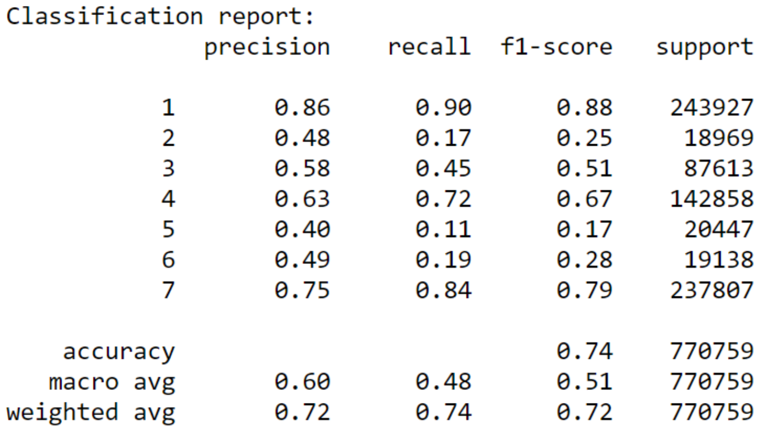
F-measure: F-measure will consider both recall and precision rate to provide a merged score.

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

4 Results

For hill climbing, the order of input the value into the model is y, z and x-axis and the score with incremental features increased from 0.38, 0.58 to 0.71. Thus, those three data features will be considered.

For decision tree, after a few iteration to narrow the number of range in different parameters, those parameters and iteration check range are max_depth_DT = range(2, 20), min_samples_split_DT = range(2,20), min_samples_leaf_DT = range(2,10) with 'gini' and 'entropy'. The results shown that The highest prediction rate is: 0.74 and relevant parameters are criterion = entropy; max_depth = 16; min_samples_split = 19; min_samples_leaf = 7. The classification report is:



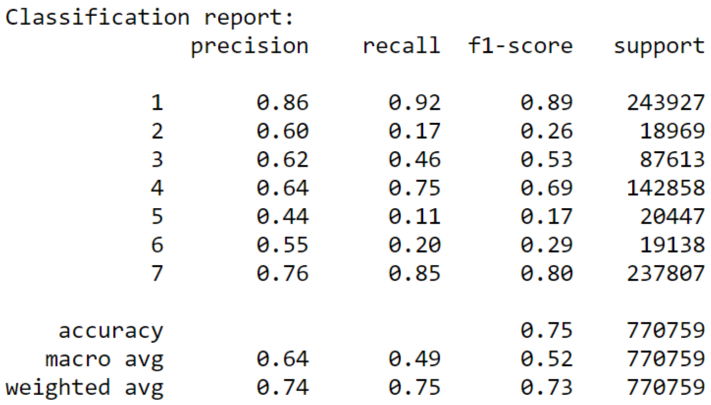
```
Classification report:
              precision    recall  f1-score   support

           1       0.86      0.90      0.88    243927
           2       0.48      0.17      0.25     18969
           3       0.58      0.45      0.51     87613
           4       0.63      0.72      0.67    142858
           5       0.40      0.11      0.17     20447
           6       0.49      0.19      0.28     19138
           7       0.75      0.84      0.79    237807

    accuracy                           0.74    770759
   macro avg       0.60      0.48      0.51    770759
weighted avg       0.72      0.74      0.72    770759
```

Figure 4. Classification Report for DT

For the KNN, those parameters range have been narrowed to n_neighbors_KNN = range(49,90,2), weights_KNN = 'uniform' and 'distance', p_KNN = range(1, 3). After three nested loop checked, the best results is: 0.753 with parameters: n_neighbors = 67; weights = uniform. Here, it needs to notice that one iteration way includes the distance and the value of p. Otherwise, it only considered weights = 'uniform'. The classification report is:



```
Classification report:
              precision    recall  f1-score   support

           1       0.86      0.92      0.89    243927
           2       0.60      0.17      0.26     18969
           3       0.62      0.46      0.53     87613
           4       0.64      0.75      0.69    142858
           5       0.44      0.11      0.17     20447
           6       0.55      0.20      0.29     19138
           7       0.76      0.85      0.80    237807

    accuracy                           0.75    770759
   macro avg       0.64      0.49      0.52    770759
weighted avg       0.74      0.75      0.73    770759
```

Figure 5. Classification Report for KNN

Both algorithms were tested by 3-Fold cross-validation because this dataset is enormous. The limited number of K is not only enough to train the model, but also decrease the computational burden. Moreover, the classification report was generated by 0.4% test data size.

5 Discussion

From the results, it is clear to see that standing up, walking and going up/down stairs, going up/down stairs and walking and talking with someone (labels 2, 5, 6) have a very low recall rate. The main reason is probably unbalanced data. From Figure 3, it gives information about only 2%, 3% and 2% proportion compared with the whole dataset, which refers to classification problems. Most machine learning classification algorithms are sensitive to unbalance in the predictor classes because they are designed to maximize accuracy and reduce error. Many popular methods can deal with the class imbalance:

- Change the performance metric:
  Use confusion matrix and classification report to judge the model more comprehensively (F1: Score). However, the score of this method does not look so impressive.
- Different algorithms:
  Usually, the decision tree, the random forest can be especially beneficial with imbalanced datasets, which worked by learning a hierarchy if/else questions and force both classes to be addressed
- Resampling techniques:
  Oversampling can add more copies of the minority class. However, if the dataset is huge, it will increase the computing burden and change the original data pattern.
  Undersampling can remove some observations of the majority class. However, it may remove valuable information, which could lead to underfitting and poor generalization.

Moreover, the reference paper for this dataset only generate five labels data [1], which is walking, stairs, talking, standing and working. That minority dataset is probably an intermediate state data that contains characteristic from other labels. Thus, this may also lead modifier to learn appropriately.

Furthermore, with the vast dataset, it is hard to explore and search the best optimal parameters to arrive at the highest accuracy rate. Parameters adjustment need a lot of computing resource. In this report, the accuracy rate is a locally optimal solution.

6 Advanced Topic

Grid search cross-validation (GrideSearchCV) with different algorithms is another right way to tune the parameters without manually process. However, there is a little bit different in the cross-validation part. In the GrideSearchCV, the cross-validation split data into training data, validation data and test data, and only K-Fold will be implemented in the training and validation data, for example, five folders. For the test data part, it is never changed and used for training in any iteration of creating modifier. For the validation data, it will mainly be used in training data with tuning parameters to check whether there are underfitting and overfitting.

The results that obtain from GrideSearchCV:

For DT: Test set score: 0.75; The best score on train set: 0.75; The best parameters: {'criterion': 'entropy', 'max_depth': 25, 'min_samples_leaf': 45, 'min_samples_split': 80}

For KNN: Test set score0.75, The best parameters:{'n_neighbors': 57, 'p': 2, 'weights': 'uniform'}

The best score on train set:0.75

It obtains a little bit increase compared with manually tuning the parameters.

Random forest is another good solution for this question from ensemble learning. This bagging algorithm will let each decision tree to judge and classify. Each decision tree will have a result of classification, and then, vote the results. The higher voter will be selected as the final results. In the dataset of accelerometer, the main drawback is that it needs extensive calculations. Thus, in the code part, it did not process further. However, Pierluigi et al. claim that the prediction rate could arrive 94% with five labels data.

7 Conclusion

The seven activities of activity recognition from single Chest-Mounted Accelerometer was predicted by two classification algorithms, KNN and DT. After feature selection, x, y, z acceleration have a positive contribution as training features by the hill-climbing algorithm. From the results, it can be seen that both KNN and DT have similar performance that is about 0.75, which may be influenced by unbalanced classes. GridsearchCV algorithm can bring a little bit higher performance in the search globality, but it requires many computing resource costs.

Reference

[1] Casale, P., Pujol, O. and Radeva, P., 2011, June. Human activity recognition from accelerometer data using a wearable device. In Iberian Conference on Pattern Recognition and Image Analysis (pp. 289-296). Springer, Berlin, Heidelberg.

[2] UCI. 2020. Activity Recognition from Single Chest-Mounted Accelerometer Data Set. [ONLINE] Available at: https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer#. [Accessed 1 June 2020].

[3] Steelcase. 2020. Beta Testing For New Ways of Sitting. [ONLINE] Available at: https://www.steelcase.com/research/articles/topics/technology/beta-testing-new-sitting-experience-gesture/. [Accessed 2 June 2020].

[4] Massaad, F., Lejeune, T.M. and Detrembleur, C., 2007. The up and down bobbing of human walking: a compromise between muscle work and efficiency. The Journal of physiology, 582(2), pp.789-799.

[5] VIRTUALSPEECH. 2020. 8 Elements of Confident Body Language. [ONLINE] Available at: https://virtualspeech.com/blog/8-elements-of-confident-body-language#:~:text=When%20you%20speak%2C%20you%20don,behind%20what%20you%20are%20saying.. [Accessed 7 June 2020].

[6] TOASTMASTERS international. 2020. Leg Posture Reveals Our Mind's Intent. [ONLINE] Available at: https://westsidetoastmasters.com/resources/book_of_body_language/chap10.html. [Accessed 7 June 2020].

[7] Machine Learning Mastery. 2020. An Introduction to Feature Selection. [ONLINE] Available at: https://machinelearningmastery.com/an-introduction-to-feature-selection/. [Accessed 8 June 2020].

[8] sklearn. 2020. sklearn.neighbors.KNeighborsClassifier. [ONLINE] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. [Accessed 8 June 2020].