Physical Activity Recognition using Chest Mounted Accelerometer

Nischay Bikram Thapa

S3819491

Master of Data Science, RMIT University

s3819491@student.rmit.edu.au

2/06/2020

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*.

Table of Contents

Abstract

Physical activity recognition is a comprehensive study that intends to identify a person's actions based on sensor data. In this paper, activity recognition is expressed as a classification task where two traditional machine learning algorithms are analysed based on their precision and accuracy. Extra features are also created to examine the impact on the model's performance. K Nearest Neighbour is found to perform consistently well in identifying distinct activities.

Introduction

Human Activity Recognition is a comprehensive field of study focusing on identifying an individual's actual movement or activity based on sensor data. Movements can be actions such as working on a computer, standing, walking, going upstairs, etc. In many pieces of research, activity recognition has been based on classifying sensor data using one or more accelerometer. [1] Several articles were also published, where certain accelerometer data analysis has been implemented and investigated for the identification of physical activity.

The goal of this paper is to choose the best model using classical machine learning techniques to solve the multi-class classification problem that answers "what is a person doing based on their behaviour?" question. To solve this problem, K-Nearest Neighbour and Decision Tree Classifiers are analysed and the best one is selected based on performance and accuracy.

The dataset was acquired from a wearable accelerometer placed on the chest that poses challenges for people using motion patterns to authenticate themselves. The problem is framed as a classification task using uncalibrated accelerometer data to distinctly identify seven different activities among 15 participants [1]. The sampling frequency recorded of the accelerometer is 52Hz with data consisting of x-acceleration, y-acceleration and z-acceleration and activity labels. Seven activities like working at a computer, standing up, walking and going up downstairs, standing, walking, going up downstairs, walking and talking with someone and talking while standing are considered to show its performance and robustness. The dataset is open sourced and publicly available at UCI Machine Learning Repository.

The paper is divided into four sections. The first section describes the methods used in data preparation, exploration and modelling for the aforementioned problem. Subsequently, the second section incorporates results that are obtained from the classification task using two machine learning models. Finally, in the other two sections, comparison between the classification algorithms, the importance of features and tuning of hyperparameter is discussed and concluded with a recommendation for further automation.

Methodology

The dataset was collected from 15 participants performing 7 activities and was available through 15 CSV files. As this is an observational study, every data file was combined into one that originally consists of 1923177 observations and 5 columns. [1] The columns represent a sequential number, x-acceleration,y-acceleration, z-acceleration and activity labels. The activity labels here are coded as integer values ranging from 1 to 7 and information for each label were defined in the source. For the ease of analysis and better visual interpretation, data for only one participant was taken as a sample. However, data modelling was done on the entire dataset. This section is further divided into three sections. The first includes methods used for data retrieving and preparation, second describes processes and insights from data exploration and the final involves data modelling and evaluation.

Data Preparation

The primary objective of this step is to retrieve the data from the source and prepare it further for analysis. Usually, raw data can have many anomalies such as missing values, typo errors, whitespaces and impossible values. Examining them at an early stage aids us to achieve accurate results and improves the quality of the data [3]. The initial step taken here was to check the data types, missing values, any data-entry errors and impossible values. While working through this step, an impossible value coded as '0' was identified and removed from the data. After following these steps, the data was well structured, clean and ready for exploration.

Data Exploration

After the data is structured, the fundamental step is to calculate descriptive statistics to summarise the feature column and understand the characteristics. The graphical technique was used to explore the column as it helps to provide deeper insights compared to numerical values. This step is further segmented into two parts:

Univariate Analysis

The data consists of three feature columns and one target label. While the feature column represents an acceleration in three axes, those were expressed using histograms and boxplot to understand the distribution and five-number summary.

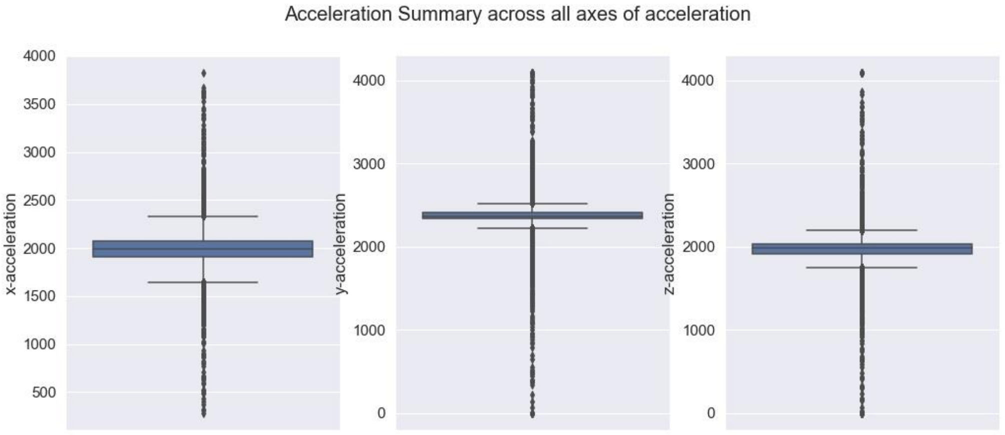


Fig 1: Boxplots for all three axes

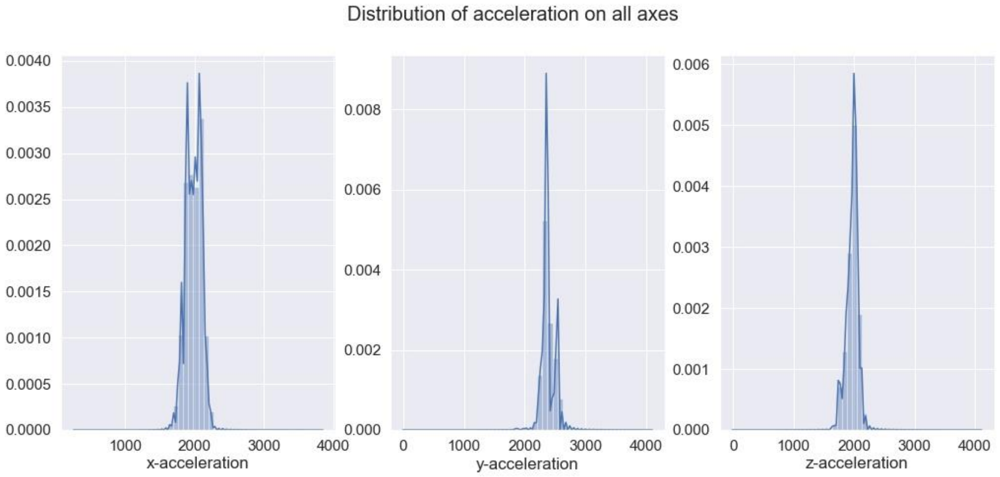


Fig 2: Histogram for all three axes

The above boxplots and histogram explain that most of the values for x-axis are in the range from 1500 to 2200. The median scale for both x and z-axis are the same whereas the median for the y-axis is comparatively greater lying in the range from 2100 to 3000. Outliers can also be seen but there was no strong evidence and hence were ignored for this analysis. Looking at the distribution, the data is tightly centred with a long tail on both sides. This unquestionably explains that most of the acceleration values lie in that shaded region.

After exploring all the numerical features individually, activity labels are plotted using a bar diagram showing the percentage of each class label across the entire dataset. From Fig 3, it is evident that 62% of activities were 1 (Sitting on a computer) and 7 (Talking while Standing) followed by others showing imbalances among the class labels. Therefore, the performance measure of this project cannot be based on accuracy but rather should be evaluated based on precision and recall.

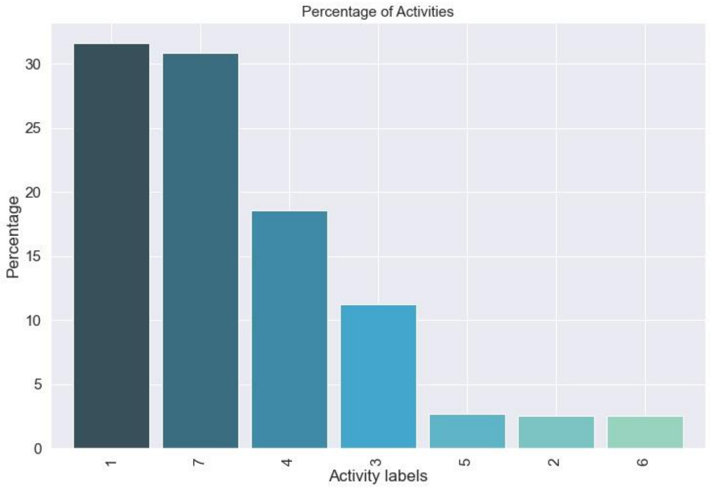


Fig 3. Percentage of Activity labels

Multivariate Analysis
In order to identify relationship between feature columns and labels, various graphical representation were used along with hypothesis test.

Hypothesis: X-axis reading for all acitivities are different.

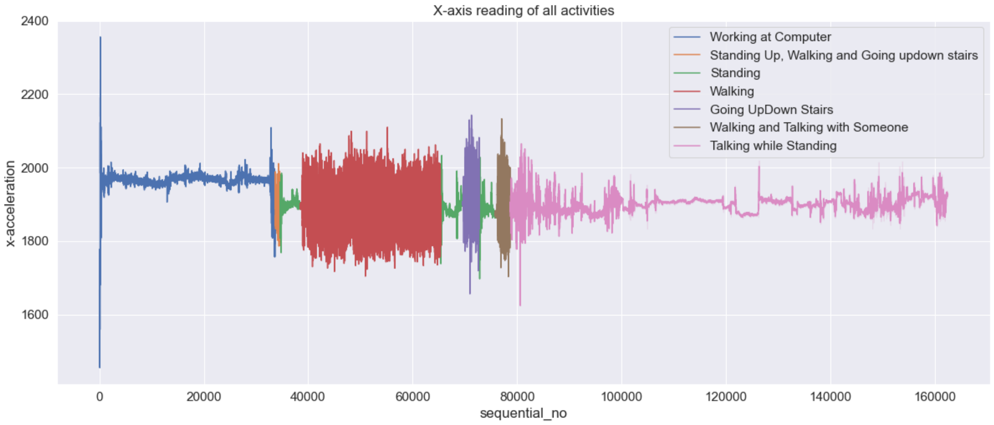


Fig 4. X-axis reading of all activities

Looking at the graph, it is evident that x-axis reading for all activities are different with each acitivity having a distinct pattern.

Hypothesis: Vertical movement in activites have higher y-axis acceleration.

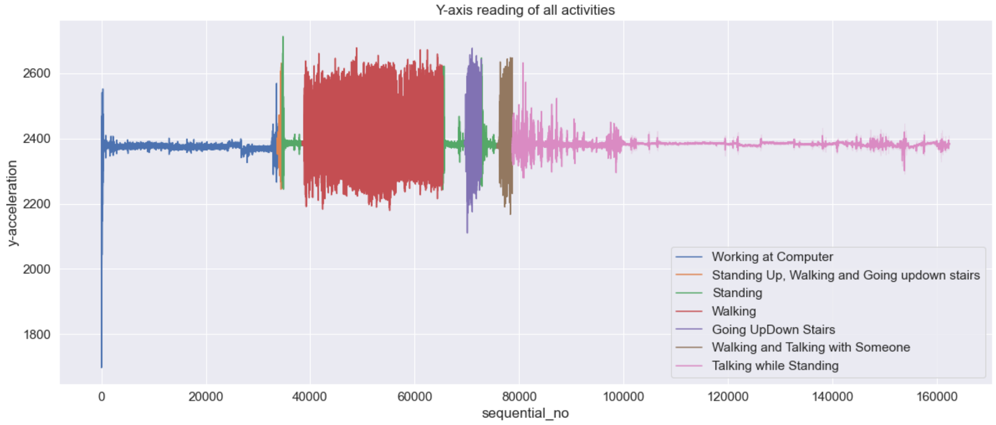


Fig. 5. Y-axis reading for all activities

From the graph, it is clear that activities such as standing,going up down stairs have slightly higher y-acceleration compared to other activities.

Hypothesis: The average acceleration of all the activities across all three axes are distinct.

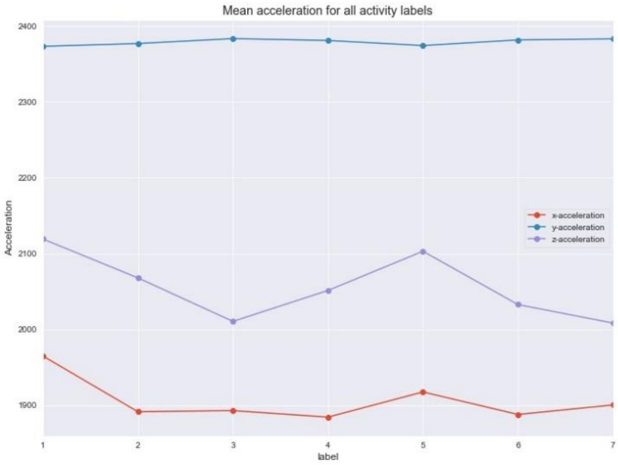


Fig.6 Mean acceleration for all three axes

The line plot provides evidence that the mean accleration for every physical activity is different with acitivity 5 having the highest average acceleration across x and z-axis. A positive correlation can also be observed between these two axes.

Hypothesis: X axis and Z axis are show a positive relationship

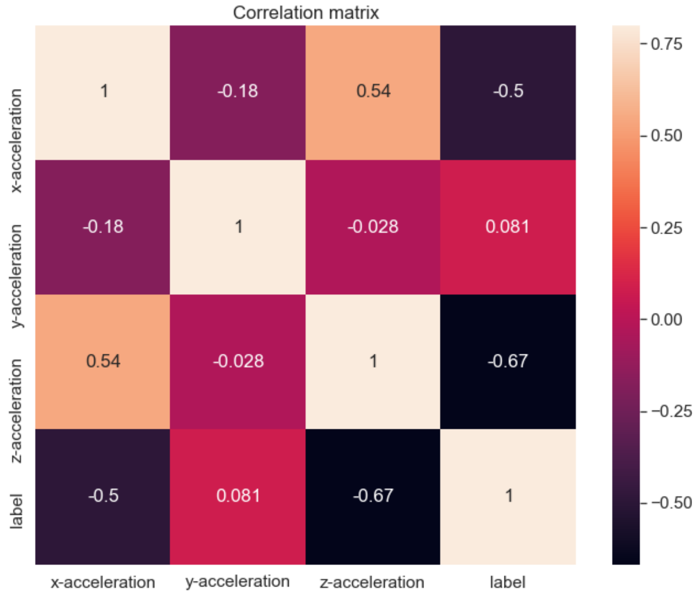


Fig.7 Heatmap illustrating correlation between attributes

From the correaltion matrix, it is evident that x-axis and z-axis are positively correlated with a coefficiant of 0.54. Similary a neagetive correaltion can be observed between y-axis and other axes.

Data Modeling

The goal of this project is to distinctly identify each of the activity by the patterns observed from acceleration across all axes. Since this is a classification task, two conventional algorithms; Decision Tree Classifier and K Nearest Neighbour Classifier are used independently to train the model. The modelling process is further divided into multiple steps and follows an iterative cycle:

Train and Test Split

First, the dataset is divided into features and class label stored in variables such as X and y. Then, both data and class variables have been split up into three pairs of test and train data sets. Dividing the dataset into train and test set helps to avoid overfitting and underfitting. The pair consists of 80% train data and 20% test data.

Feature Selection

From the above data exploration, it is clear that there are significant patterns observed for each activity from acceleration on all axes. Hence, these are considered as the features for the data that can be used to predict the target label.

Model Selection

The Supervised Machine Learning technique requires human interaction to label the data. In this technique, patterns can be found from the previous outcomes of observations which are learnt by the model and are used to find the outcome of new observations. Classification, a type of supervised learning approach is used to identify unseen activities from previously recognized activities. For this purpose, Decision Tree Classifier and K Nearest Neighbour are used, compared and the best one is selected for future purposes.

Decision Tree

The decision tree classification algorithm can work on both continuous or categorical features. Decision tree splits the observations based on the most effective splitter which splits the observations into two or more homogeneous groups of observations. Each decision tree consists of three important components: an internal node which represents a test condition, a leaf node which represents the class label and branch which represents the result of the test.

K-Nearest Neighbour

The K-Nearest Neighbour classification algorithm classifies the new observation based on the k closest training observation in the model. The parameters used to determine the optimal nearest neighbour depends upon selecting the number of neighbours, weights assigned to each data point and the metric used to calculate the distance between the data points.

Model Training

The data obtained after the train test split along with the modelling techniques are used to train the model. This step is presented with the split pair of data from which it can learn using two different classification approaches. As a result 4 trained model was generated, then compared with the performance metric and the optimal one is used for further tuning.

Model Evaluation

After successfully training the models, it is evaluated using different performance measure such as accuracy, precision and recall score. As the dataset is imbalanced, the priority is given to models that hold a good precision score i.e the model is trustworthy in the class labels it predicts and good recall i.e the ability of the model to detect the class label. However, due to similar activities such as standing and talking while standing, a good precision score can be considered to evaluate the model's performance.

Feature Engineering

After creating a baseline training model, new features such as the magnitude of all three acceleration and correlation between each attribute are added to improve the performance. Every activity forms a distinct pattern that is evident in Fig 4, 5 and 6. For instance, activity such as walking differs significantly from climbing upstairs due to different acceleration intensity along different axes. Hence, creating a new feature with the distance can aid to distinctly identify physical activities. The magnitude of the accelerations can be calculated as $\sqrt(x^2 + y^2 + z^2)$ [1]. Similarly, different acceleration along the axes can have a positive or negative influence in identifying the activities [4]. For instance, when a person is moving along the x-axis or horizontally, their y-axis acceleration or vertical direction is minimum or close to zero. These kinds of relationship between each attribute can also help to determine physical activities to some extent and thus are added to the model.

Hyperparameter Tuning

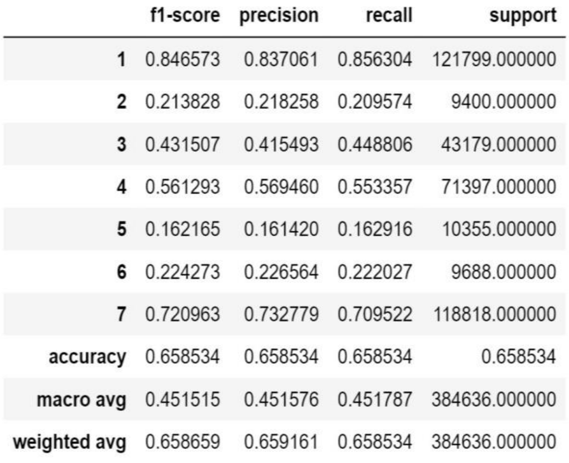
The best optimal model is selected and the parameters are tuned such that the model effectively learns from the data and the results are improved from the baseline. The parameters of KNearest Neighbours such as nearest neighbours, distance metric and weights are tuned to obtain optimal results.

Stratified K Fold Cross-Validation

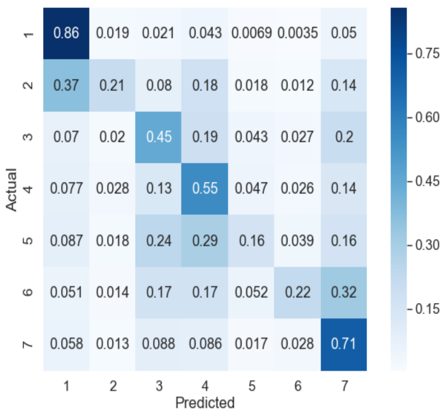
Stratified K Fold Cross-validation is a resampling strategy used to evaluate the model performance by splitting the data into k number of folds and groups known as strata. The dataset is shuffled randomly and divided into k groups where a certain percentage of stratified data are used for training and testing. Although the model results in good accuracy with one fold, cross-validation helps to generalize from the results obtained.

Results

Baseline Decision Tree Classifier



	f1-score	precision	recall	support
1	0.846573	0.837061	0.856304	121799.000000
2	0.213828	0.218258	0.209574	9400.000000
3	0.431507	0.415493	0.448806	43179.000000
4	0.561293	0.569460	0.553357	71397.000000
5	0.162165	0.161420	0.162916	10355.000000
6	0.224273	0.226564	0.222027	9688.000000
7	0.720963	0.732779	0.709522	118818.000000
accuracy	0.658534	0.658534	0.658534	0.658534
macro avg	0.451515	0.451576	0.451787	384636.000000
weighted avg	0.658659	0.659161	0.658534	384636.000000

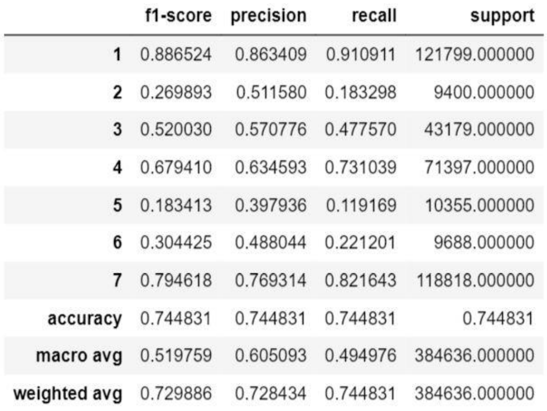


Actual \ Predicted	1	2	3	4	5	6	7
1	0.86	0.019	0.021	0.043	0.0069	0.0035	0.05
2	0.37	0.21	0.08	0.18	0.018	0.012	0.14
3	0.07	0.02	0.45	0.19	0.043	0.027	0.2
4	0.077	0.028	0.13	0.55	0.047	0.026	0.14
5	0.087	0.018	0.24	0.29	0.16	0.039	0.16
6	0.051	0.014	0.17	0.17	0.052	0.22	0.32
7	0.058	0.013	0.088	0.086	0.017	0.028	0.71

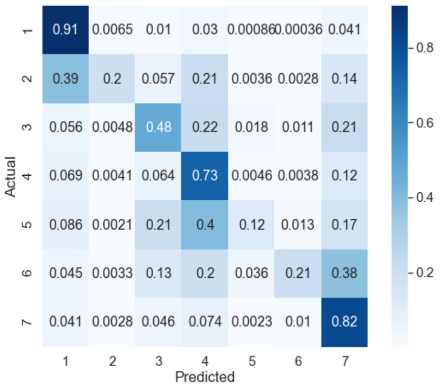
Fig 8. Classification report and confusion matrix obtained from descion tree classifier

Decision Tree Classifier results in 65.8% accuracy where it can predict activities such as 1 and 7 very well. However, it has low precision and recall score for other activity labels.

Baseline K Nearest Neighbour



	f1-score	precision	recall	support
1	0.886524	0.863409	0.910911	121799.000000
2	0.269893	0.511580	0.183298	9400.000000
3	0.520030	0.570776	0.477570	43179.000000
4	0.679410	0.634593	0.731039	71397.000000
5	0.183413	0.397936	0.119169	10355.000000
6	0.304425	0.488044	0.221201	9688.000000
7	0.794618	0.769314	0.821643	118818.000000
accuracy	0.744831	0.744831	0.744831	0.744831
macro avg	0.519759	0.605093	0.494976	384636.000000
weighted avg	0.729886	0.728434	0.744831	384636.000000



Actual \ Predicted	1	2	3	4	5	6	7
1	0.91	0.0065	0.01	0.03	0.00086	0.00036	0.041
2	0.39	0.2	0.057	0.21	0.0036	0.0028	0.14
3	0.056	0.0048	0.48	0.22	0.018	0.011	0.21
4	0.069	0.0041	0.064	0.73	0.0046	0.0038	0.12
5	0.086	0.0021	0.21	0.4	0.12	0.013	0.17
6	0.045	0.0033	0.13	0.2	0.036	0.21	0.38
7	0.041	0.0028	0.046	0.074	0.0023	0.01	0.82

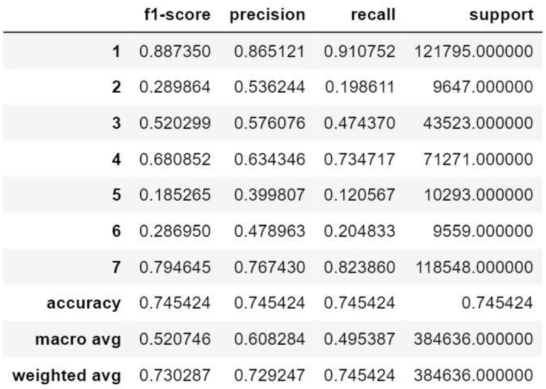
Fig 8. Classification report and confusion matrix obtained from K nearest neighbour classifier

The accuracy score of K-Nearest Neighbour classifier obtained with default parameters is 74%. In addition to this, the f1-score of predicting activity labels is better than Decision Tree.

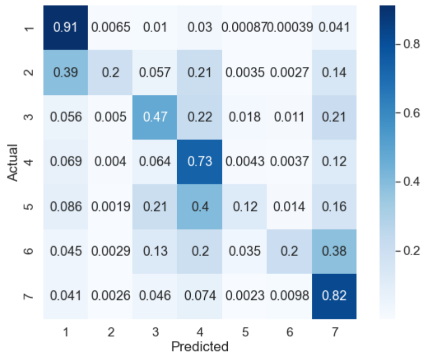
Model Performance after Feature Engineering

Comparatively, KNN performed better, this model is selected as new features are added to improve the performance. Optimal Hyperparameters such as the number of neighbours, weights and other parameters are compared and the best parameters with 13 number of neighbours, uniform weight, and Minkowski distance are used. The results exhibit a slight improvement in predicting some class labels than the baseline model.

K-Nearest Neighbour



	f1-score	precision	recall	support
1	0.887350	0.865121	0.910752	121795.000000
2	0.289864	0.536244	0.198611	9647.000000
3	0.520299	0.576076	0.474370	43523.000000
4	0.680852	0.634346	0.734717	71271.000000
5	0.185265	0.399807	0.120567	10293.000000
6	0.286950	0.478963	0.204833	9559.000000
7	0.794645	0.767430	0.823860	118548.000000
accuracy	0.745424	0.745424	0.745424	0.745424
macro avg	0.520746	0.608284	0.495387	384636.000000
weighted avg	0.730287	0.729247	0.745424	384636.000000



Actual \ Predicted	1	2	3	4	5	6	7
1	0.91	0.0065	0.01	0.03	0.00087	0.00039	0.041
2	0.39	0.2	0.057	0.21	0.0035	0.0027	0.14
3	0.056	0.005	0.47	0.22	0.018	0.011	0.21
4	0.069	0.004	0.064	0.73	0.0043	0.0037	0.12
5	0.086	0.0019	0.21	0.4	0.12	0.014	0.16
6	0.045	0.0029	0.13	0.2	0.035	0.2	0.38
7	0.041	0.0026	0.046	0.074	0.0023	0.0098	0.82

Fig 8. Classification report and confusion matrix obtained from K nearest neighbour classifier after feature engineeering

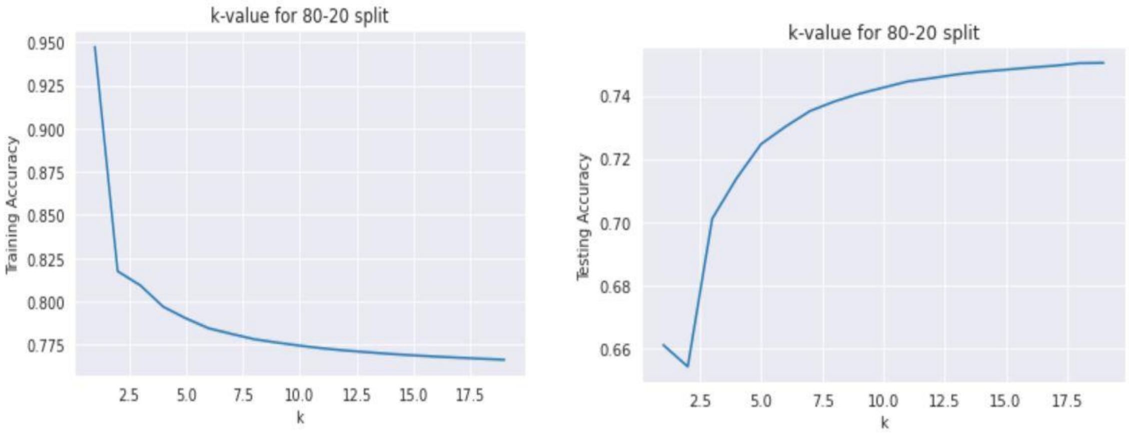


Fig 9. Traning and Testing Accuracy of K Nearest Neighbour

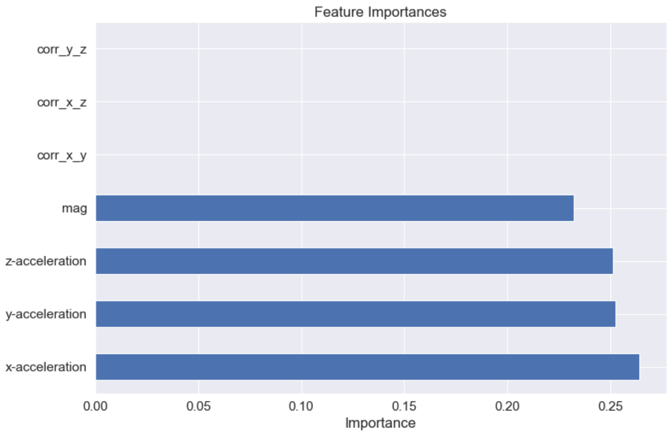


Fig 10. Feature Importances

Discussion

In overall, it is evident that the dataset is imbalanced which suggests that we can't solely rely on accuracy as the performance metric to determine the best model for this dataset. While precision is the ratio of correctly predicted positive observations of the total predictive positive observation, this measure is taken into consideration. Classification models such as Decision Tree and K Nearest Neighbours are used to learn and predict activity labels from the acceleration recorded in three different axes. From the above results, it is clear that K Nearest Neighbour Classifier performed better in predicting the class labels compared to the decision tree based on the precision or overall f1-score. Hence, this model is selected for further tuning and finding the optimal parameter to generalize beyond the dataset. New features were also added to see whether they influence the performance of the model and the result showed that the new feature 'magnitude' was able to improve the precision score of some target. Hyperparameters such as the number of neighbours, weights and distance function were checked manually and the best parameters were identified to improve the overall weighted score. Moreover, stratified 5 fold cross-validation was produced from the entire dataset that resulted in an average accuracy score of 74.6%. This shows the model was able to learn and generalize from the data. However, a poor precision and recall score still can be noticed due to the imbalances in class labels where activity 1 and 7 were in majority compared to other activities. If an approach to balance the class label was considered, the model would be able to perform much better with a significant rise in f1-score.

Conclusion

In conclusion, K Nearest Neighbour with 13 number of neighbours having uniform weight and selecting the Minkowski distance with features such as x-acceleration, y-acceleration, z-acceleration and magnitude of x,y and z behaved precisely better in predicting the physical activities. However, due to imbalance in the dataset, the model could not perform better but can be used to distinctly identify activities such as working on a computer, talking while standing and walking. There were similar activities such as standing up, walking and going updown stairs, going updown stairs, walking and talking to someone which the model could not differentiate well. If these activities were grouped and classes were balanced, this model could perform much better and used in production for future purposes.

References

[1] P. Casale, O. Pujol and P. Radeva, "Human Activity Recognition from Accelerometer Data Using a Wearable Device," Barcelona, 2011.

[2] P. Casale, O. Pujol and P. Radeva, "Personalization and user verification in wearable systems using biometric walking patterns Personal and Ubiquitous Computing," 2011.

[3] Y. Ren, "Data Curation," in *Practical Data Science with Python*, RMIT University, 2020, pp. 1-57.

[4] N. Ravi, N. Dandekar, P. Mysore and M. L. Littman, "Activity Recognition from Accelerometer Data," in *Rutgers University*, Piscataway, 2005.