RMIT UNIVERSITY

HUMAN ACTIVITY RECOGNITION

Building classification models for Recognizing the Human Activities based on the Accelerometer Axial Data

KARTHI NARENDRABABU GEETHA
S3835901
RMIT University

Table Of Contents

1. ABSTRACT:

The scope of this report is that, the Human Activity Recognition will be demonstrated with the help of 2 Classification models which is K-Nearest Neighbour and the Decision tree classification. The Human activities are going to be classified with help of recorded Accelerometer's axial data.

2. INTRODUCTION:

As the technology improving in the field of Ubiquitous Computing, the researchers are interested in recognizing the Human Activities. Human Activity Recognition is one of the blooming technologies in the field of Pervasive Computing, where the actions of human's everyday life activities are identified.

STEPS FOR ACTIVITY RECOGNITION PROCESS:

Step-1: Initially the activities of human are recorded with the help of Accelerometer.

Step-2: In the next step, the collected raw sensor data are pre-processed, by eliminating the noise, data redundancy and handling the data incompleteness.

Step-3: Identifying the relevant data sets for recognizing the human activities.

Step-4: Extracting the correct features from the segmented data set in order to proceed with classification.

Step-5: Classifying the human activities with the help of extracted features from the data segment.

3. METHODOLOGY:

a. DATA ACQUISITION:

The data for the Activity recognition process is collected from the fourteen volunteers which includes 3 women and 11 men within the age 27 and 35. Volunteers were asked to start the recognition system when they start an activity and restart the system when they perform another activity, based on which the activities were labelled. The collected data set consists of the activities performed by the volunteers such as walking up/down the stairs, walking, talking, staying standing and working at computer.

b. DATA PREPARATION:

As the dataset is composed by the activities of each human, the dataset is combined and loaded into the Python platform. The collected data set consists of about 1926896 instances with 5 features including ID of Accelerometer, Directional data (X_Axis, Y_Axis, Z_Axis), the activities that is performed by the humans (0, 1, 2, 3, 4, 5, 6, 7) and the ID of each person to differentiate between the subjects. It is clearly explained that the activities are labelled as follows,

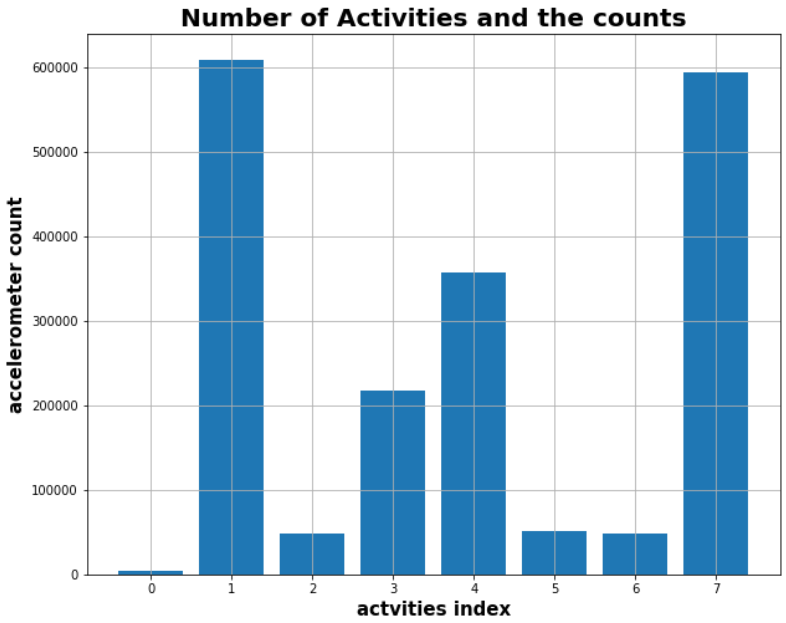
1. Working at Computer
2. Standing Up, Walking and Going up/down stairs
3. Standing
4. Walking
5. Going Up/Down Stairs
6. Walking and Talking with Someone
7. Talking while Standing

Since there is no activity assigned for the label 0, the data instances with label 0 were removed, considering that they are outliers.

c. DATA EXPLORATION:

i. IMBALANCED DATA:

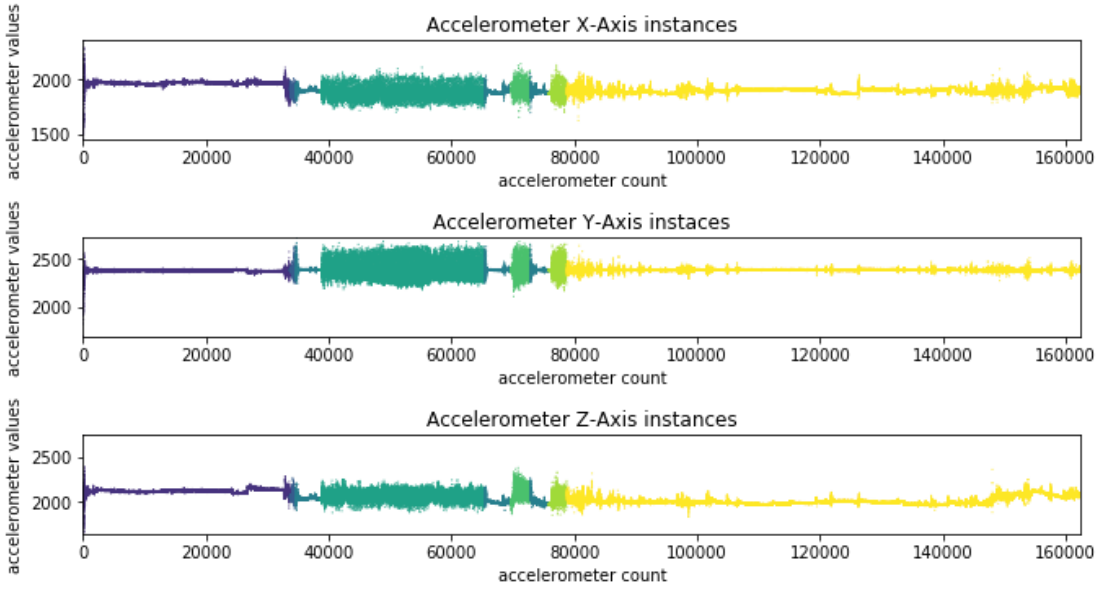
As there are 7 classes in the target data, but the number of data instances for each class is imbalanced. The below graph clearly shows that the activity 1 and 7 has increased number of data instances, whereas the activity 2, 5 and 6 has much lesser amount of data. Due to the data imbalance, the precision rate of the model prediction can be reduced, i.e. that the classification model will classify the Activity class 7 instead of class 2.



The graph also shows an extra activity class as discussed before, which can be considered as an outlier. Hence the class 0, which is contained of about 3719 data instances has been removed.

ii. EXPLORING THE ACCELEROMETER INSTANCES:

As there are 15 volunteers, who were asked to perform the activities for the purpose of data collection. Then the loaded dataset is split, which contains the data for an individual. The data that is evidently seen from the graph shows that the data collection is at the peak, during the initial stages.

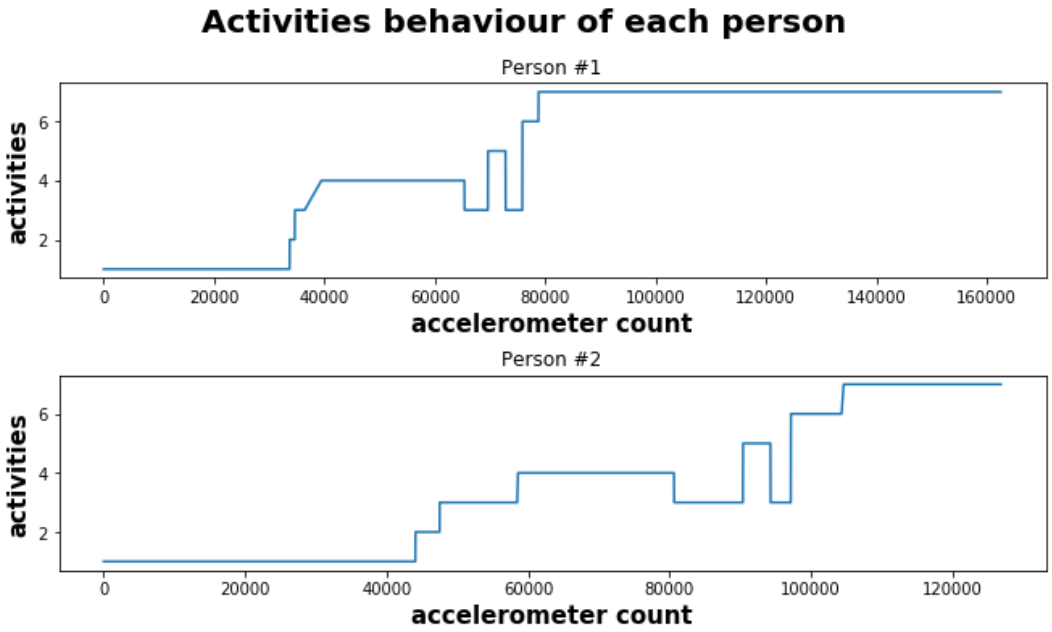


There are massiSCIve fluctuations which can be observed in the graph, as the individual performs different activities at different point of time.

In order to label the activities, the volunteers were asked to perform different activities in sequential order and restarts the system, once the activity is completed. The system takes about 2 minutes of time to initialize when the volunteer restarts it. This clearly explains the initial peaks in the above graph.

iii. EXPLORING THE ACTIVITIES OF EACH VOLUNTEERS:

The accelerometer will starts sensing the velocity of the activities performed by the volunteers, when they press the start button. From the below chart, it can be seen that, each volunteer performing different activities in different pattern. For example, for the Person#1, the activities were labelled as 0, 1, 2, 3, 4, 5, 6, 7. Similarly for the other volunteers as well.



On taking a closer look on the chart, it can be seen that some of the activities were repeated again and again. For the Person #1, the pattern of activities recorded by the system is like, 1, 2, 3, 4, 3, 5, 6, 7, the activity 3 has been repeated twice.

Then the pattern of activities for another person is like, 1, 2, 3, 4, 3, 5, 3, 6, 7 and similarly, the activity 3 has repeated twice. This shows that the precision of data recorded by the accelerometer is almost similar for all the volunteers.

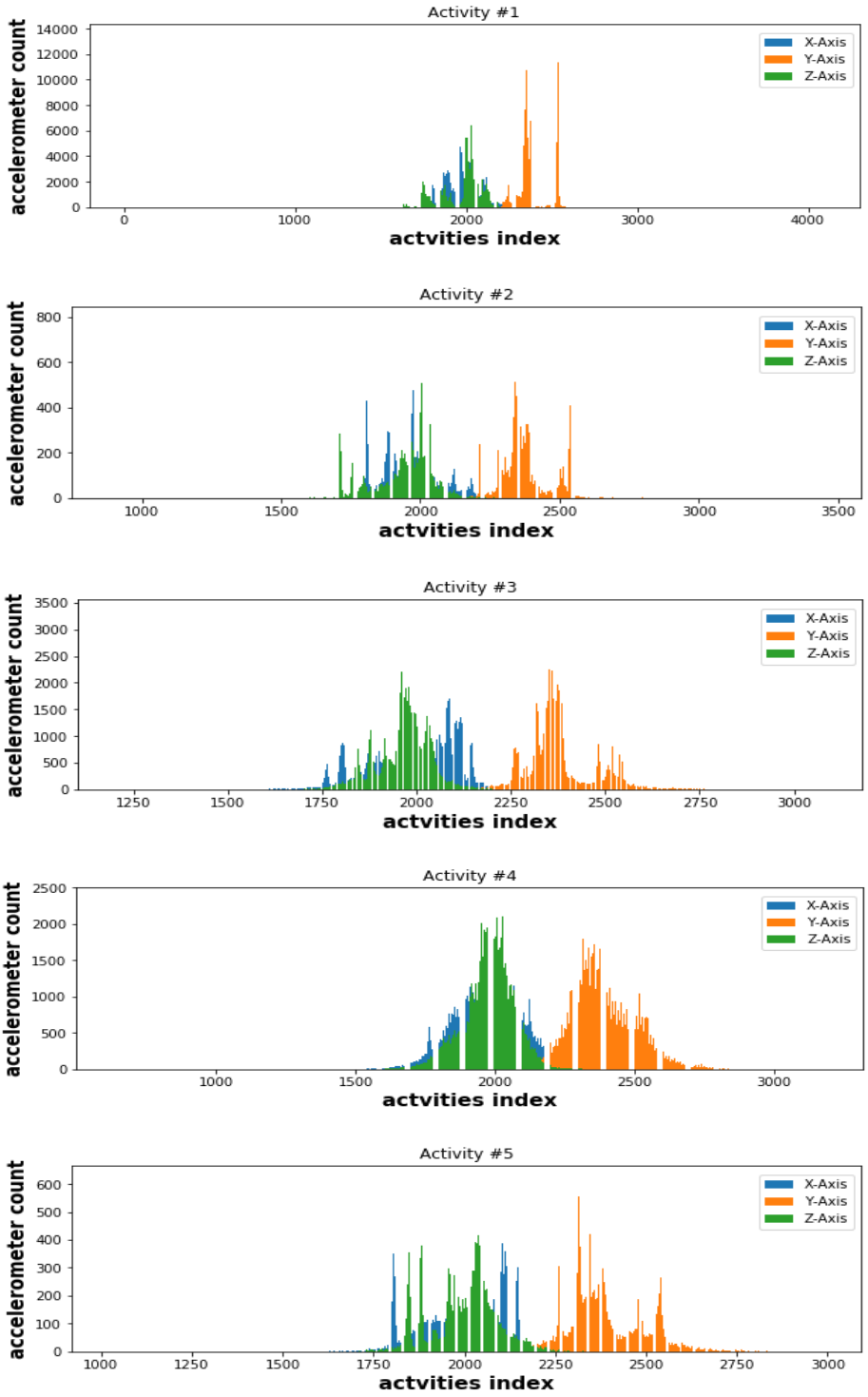
Although the activity pattern is similar for every volunteer, the chart shows that, the volunteers has taken much longer time for some of the activities than the others. For example, the person #1 performed the activity 7 (Talking while Standing) for a very longer time, while the person #2 performed activity 1 (Working at the Computer) for a very long time.

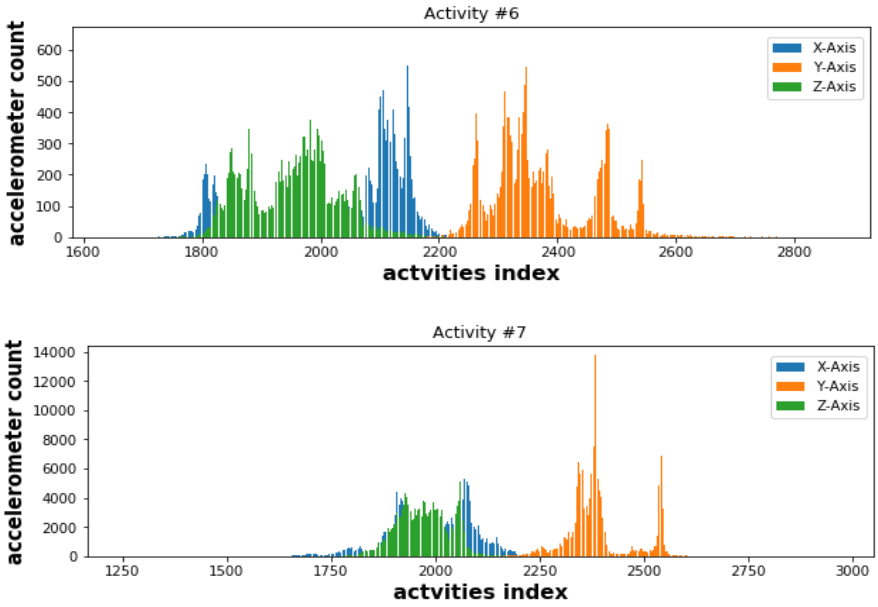
Since the activities were repeated for each volunteer, the output of the accelerometer was found to be similar in those activities which determines the accuracy of the accelerometer output.

Due to this pattern and recording time, the capability of classification model is being affected, as the records for the activities 2, 5, 6 were very low and the model might classify the activities incorrectly.

iv. EXPLORING THE X, Y AND Z AXIS OF ACCELEROMETER:

The number of data instances in the X & Z axis of accelerometer is almost same when compared to Y axis. The below chart is plotted with X, Y and Z axis of accelerometer against the total count of each axis. The chart clearly shows that the X & Z axis count is almost same, while the Y axis count is much larger.





By comparing the count of each activities, it can be seen that the activities that can be performed easily has an increased count of data instances when compared to the intense activities. For example, the activity 1 (Working at Computer) and activity 7 (Talking while Standing) are the ones which should not take much body rotation or pressure, but the recorded count of these activities is much larger than the other intense activities.

Also, the split graphs of each activities show that the, Y axis of the accelerometer excites only for the easily performed activities and because of which the data instances of those activities are higher. For the actions such as Standing Up, Walking and Going up/downstairs, the axial data count are almost same, however the number of data instances in much lesser.

d. DATA MODELLING & CLASSIFICATION:

As the scope and goal of the problem in Human Activity Recognition is explored, the next step is to select the features to proceed with the modelling.

i. FEATURE SELECTION:

As there are only 3 features, all the three features were selected. If there are more than 20 features, then the Hill Climbing feature selection algorithm can be used to select the appropriate features to proceed with the modelling. Hence the features for the classification model will be the axis coordinates of the accelerometer.

ii. SCIKIT-LEARN:

With help of Scikit-Learn, which is a machine learning library specifically developed for the python programming language, the data modelling and classification will be done. There are many features in the Scikit-Learn machine learning library and in this report, the data classification will be explained with the help of K-Nearest Neighbours classification and the Decision Tree classification algorithms.

iii. TEST, TRAIN SAMPLE SPLIT:

To proceed with building the classification model, the collected data should be split into 2, Train and Test. The classification model will be trained with the help of the training data set and the predictions are done with the help of test data.

The scikit learn library has a library named **model_selection**, which has a function named **train_test_split**. With the help of the train_test_split method, the data is split by providing the weightage for the testing data set. Now the shape of Train and Test data set will be like,

```
X_train: (1348827, 3)
 X_test: (578069, 3)
 y_train: (1348827,)
 y_test: (578069,)
```

iv. K-NEAREST NEIGHBOURS CLASSIFICATION:

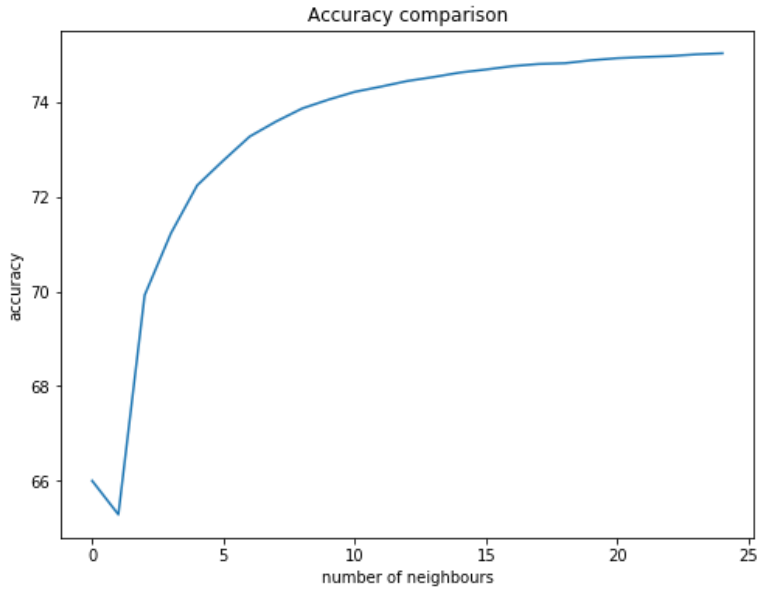
As the valid features are selected and the train, test data are acquired, the classification model needs to be developed. As discussed, the activities were classified with the help of the K-Nearest Neighbours classification algorithm which is a supervised machine learning technique.

In order to proceed with the classifications, the training data set needs to be passed to the algorithm. The fit method in the sklearn library is used to train the dataset based on the selected classification algorithm. The classification model is trained based on the taken training data set. Initially, the classification algorithm is ran without any parameters passed to it and the accuracy of predicting the test data set is about 72.24%.

1.HYPER PARAMETER TUNING:

The parameters given to the classification model will actually improve the accuracy of the classification. There 3 major parameters in the K-Nearest Neighbour classification algorithm which includes number of neighbours, metric and the power parameter value.

The power parameter should be lesser for the high dimensionality data and hence the p value will be set to 1. Since the power parameter is taken as 1, the metric is equivalent to the "**Minkowski**". Then model is tested with about 25 number of neighbours starting from 5 neighbours. The parameters are chosen based on the best accuracy of prediction, which is the metric: "Minkowski", p:1 and n_neighbours:24.



The above graph clearly depicts that the accuracy has been increased with the increase in the number of neighbours.

2.VALIDATION:

Once the hyper parameters are identified and the next step is to validate the classification model. Scikit-Learn library helps to generate a confusion matrix and the classification report for the trained model.

With the help of confusion matrix, the predicted data can be validated. The matrix is created with NxN values, where the rows will be the trained target data and the columns will be the predicted data. The number of correct hits will be recorded for each of the target classes.

```
[[158482   1010   1596   5378    115     64   7529]
 [  5661   2688    607   3091     47     25   2281]
 [  3252    229  28330  14284   1078    645  13799]
 [  7385    304   5043  79344    301    325  12985]
 [  1337     15   2926   6562   1778    211   2762]
 [   690     31   1699   2843    517   2972   5545]
 [  6930    347   6372  12316    338   1447 149055]]
```

From the above matrix, it can be seen that the model predicted activity 1 correctly for about 158482, whereas it is predicted as activity 7 for about 7529. The sum of numbers on the main diagonal is the exact equivalent for the prediction of the target classes.

	precision	recall	f1-score	support
1	0.86	0.91	0.89	174174
2	0.58	0.19	0.28	14400
3	0.61	0.46	0.52	61617
4	0.64	0.75	0.69	105687
5	0.43	0.11	0.18	15591
6	0.52	0.21	0.30	14297
7	0.77	0.84	0.80	176805
accuracy			0.75	562571
macro avg	0.63	0.50	0.52	562571
weighted avg	0.74	0.75	0.73	562571

With the help Classification report method, the precision of each target classes is identified. From the above report, precision of activity 1 is 86%, however for the activity 5 is just 43%.

v. DECISION TREE CLASSIFICATION:

Decision tree classification is another supervised machine learning technique, in which the data is split into several homogeneous sets until the classification is done. The classification model is trained based on the taken training data set. Initially, the classification algorithm is ran without any parameters passed to it and the accuracy of predicting the test data set is about 65.56%.

1. HYPER PARAMETER TUNING:

In the decision tree classifier, the important hyper parameters are the depth of the tree, maximum leaf nodes, minimum samples leaf, minimum samples split. The higher the depth of the tree, the higher the chance the model lead to overfitting of data. The maximum leaf nodes is the number of leaves that the tree can have. Minimum sample leaves is the minimum of leaves the classified tree can have. Minimum samples split defines how and when the tree can be split in to two.

At the initial train, the depth of tree is set to None and then validate with how much depth the tree has taken to classify, and it is depth 14. Hence the depth is set to looped between 10 and 11. The minimum sample leaf should be at least 2 and hence it is set to 2. The maximum leaf nodes is looped between 200 and 2000 with the multiples of 100. The maximum sample split is looped between 2 and 50 with the multiples of 10. The model is looped with the above specified values and the obtained the values which had maximum accuracy.

2. VALIDATION:

Validation is as similar as it is done with the K-Nearest Neighbours algorithm; the confusion matrix is generated.

```
[[155456    803   1294   4714     39     42  11826]
 [  6028   1522    481   2778     35      8   3548]
 [  6144     25  23413  13356   1224    389  17066]
 [ 10510    164   5211  72944    180    146  16532]
 [  1731      4   2822   6128   1485     97   3324]
 [   918      2   1423   2607    765   1698   6884]
 [  9674      9   7126  11098    218    869 147811]]
```

The confusion matrix is almost like the one which is generated for the KNN Algorithm and it can be seen that the model predicted activity 1 correctly for about 155456, whereas it is predicted as a ctivity 7 for about 11826. However, the prediction count was better in the KNN algorithm.

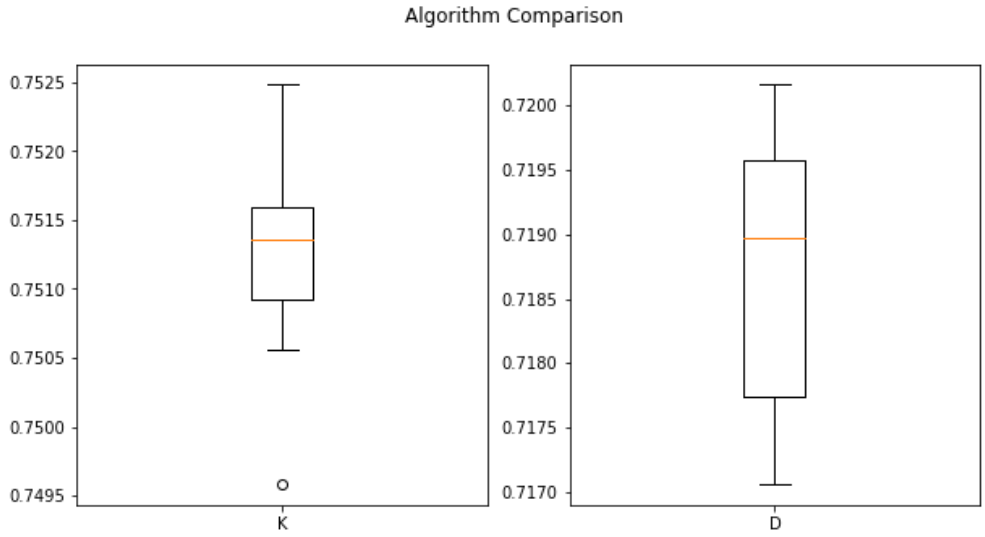
	precision	recall	f1-score	support
1	0.82	0.89	0.85	174174
2	0.60	0.11	0.18	14400
3	0.56	0.38	0.45	61617
4	0.64	0.69	0.67	105687
5	0.38	0.10	0.15	15591
6	0.52	0.12	0.19	14297
7	0.71	0.84	0.77	176805
accuracy			0.72	562571
macro avg	0.60	0.45	0.47	562571
weighted avg	0.70	0.72	0.69	562571

The precision of classifying the activity 1 is 82% and it is correctly predicted for about 89%, however the precision is much lower for the activity 5 which is only 38% and its recall rate is just 10%. The low er precision rate is due to the data imbalance.

e. COMPARISON BETWEEN THE KNN AND DECISION TREE ALGORITHM:

Once the data is trained and validated the accuracy, precision of the model, the next step is the cross v alidate the trained model with the test data. In order to proceed with the cross validation, data is split i nto K folds with the help of KFold model selection method. Each fold can actually be a test data set, th e remaining folds will be acts as a training data set. Then the cross validation can be done using the Sc ikit Learn Cross_val_score. The parameters for the Cross_val_score, cross validation method should b e the model, training data, test data, cross validation model selection which is kfold and the scoring ty pe.

With the help of Cross_val_score method, the two models KNN and Decision tree is cross validated a nd each will have 10 accuracy results, as the KFold split value is 10. On plotting the accuracy results i n a boxplot, it can be seen that the KNN algorithm is providing a little higher accuracy. Eventhough th e accuracy of decision tree model is slightly lower than the KNN model, the decision tree can handle l arger amount of data as the classification is provided based on the number of splits it is taking. Howev er, the KNN algorithm might not handle much data, as it is purely based on its neighbour data.



4. RESULTS:

On validating the trained model, it can seen that the models classified the activities correctly, for whic h the data instances is high. The KNN classification model has higher accuracy when compared to the Decision tree classification.

As discussed, the precision and recall rate for the activity classes such as 2, 3, 5, 6 is much lesser when compared to the other activities. The precision of the model can be increased if the data is in a balance d state, in order to make the data balanced, the amplification techniques or the sampling techniques ca n be used. The model will classify the activities, if it has provided the balanced and noise less data.

5. CONCLUSION:

To summarize, the Human Activity Recognition is a blooming technology and the researchers were pu tting their effort in identifying the activities of the humans. The Accelerometer data is loaded into the Python platform and the incorrect target value is removed. With the help of exploration, it was identifi ed that the provided data is imbalanced because the accelerometer axial data is higher for the easy phy sical activities and some of the activities were recorded in repetition.

All the 3 features were selected in order to proceed with the data modelling, accuracy of KNN classifi cation model was about 75%, while the accuracy of the Decision tree model is about 72%. On validati ng the model it was seen that the prediction was provided correctly for the activities with higher data i nstances. The recommendation that can be suggested here is that the recorded accelerometer data shou ld be amplified for the activities which requires more body pressure, in order to make the get the balan ced data instances.

6. REFERENCES:

1. Casale, P., Pujol, O. and Radeva, P., 2011. *Human Activity Recognition From Accelerometer Data Using A Wearable Device*. [online] Barcelona, Spain: Computer Vision Center, Bellaterra, Barcelona, SpainDept. of Applied Mathematics and Analysis, University of Barcelona, Viewed 1 June 2020 <https://www.researchgate.net/publication/221258784_Human_Activity_Recognition_from_Accelerometer_Data_Using_a_Wearable_Device/>
2. Brownlee, J., 2018. *A Gentle Introduction To A Standard Human Activity Recognition Problem*. [online] Machine Learning Mastery, Viewed 2 June 2020, <https://machinelearningmastery.com/how-to-load-and-explore-a-standard-human-activity-recognition-problem/>
3. Alzahrani1, M. and Kammoun, S., 2016. Human Activity Recognition: Challenges and Process Stages. *International Journal of Innovative Research in Computer and Communication Engineering*, [online] 4(5), Viewed 8 June 2020,<http://www.rroij.com/open-access/human-activity-recognition-challenges-and-process-stages-.pdf/>