Final Project Report
Human Activity Recognition

Suhas - 915623705
Shivang - 915623718
Sai Kopparti - 915623695

June 14, 2018

1 Data

The dataset is taken from UCI Repository named as Heterogeneity Activity Recognition Data Set. The data is characterized as Multivariate and is used for associated task like Clustering and classification.

The dataset contains the readings of two motion sensors commonly found in smartphones. Reading were recorded while users executed activities scripted in no specific order carrying smartwatches and smartphones.

1.1 Goal

We need to recognize the human activities of different user by using Classification and clustering techniques.

1.2 Total Activities

Biking, Sitting, Standing, Walking, Stair Up and Stair down.

1.3 Total Sensors

Two embedded sensors, i.e., Accelerometer and Gyroscope, sampled at the highest frequency the respective device allows.

1.4 Total Devices

4 smartwatches (2 LG watches, 2 Samsung Galaxy Gears) 8 smartphones (2 Samsung Galaxy S3 mini, 2 Samsung Galaxy S3, 2 LG Nexus 4, 2 Samsung Galaxy S+)

1.5 Used Devices

nexus_2b, nexus_1c, nexus_2c, s3_1, s3_2, s3_mini_1

1.6 Recordings

9 users data recordings but we considered only 3 users data for this project.

2 Filtering data

We got the data with 128 features which is not required for the present classification so firstly for all the below models we filtered the data such that it contains only (4 parameters) sensor outputs and its corresponding output label, and instead of data with char labels we modify the labels with integers such itâĂŹs flexible for us to do operations faster. We wrote this code in R programming and submitted in the zip file.

3 Difficulty of the proposed work

Dealing with Huge data of about 3.07 GB.

Real time data with little variation in coordinates to identify exactly into which category the test data falls in.

Dealing with various devices where there is possibility in variation of X,Y and Z coordinates data collection like the rate at which different devices recognize the variations in actions.

4 Algorithm

The clustering and classifications strategies used for the above dataset in order to identify and classify the action performed by the user are:

S No	Algorithms
1	K nearest neighbour.
2	Logistic Regression (one vs all)
3	XGBoost
4	Random Forest

5 K Nearest Neighbors:

5.1 Description

Based on the majority label of the k nearest neighbour in the training data we are predicting the output for all the test instance.

5.2 Complexity of the algorithm:

The data which we have to classify is having 3 input features to of each coordinate value and its label, by using the same parameters we trained the data (70 % of input) and tested with rest (30 % of input) when we changed the parameters by considering the time also along with 3 coordinate parameters the results are falling from 85% to 62% so we came to know that the time parameters has not much importance because it makes sense because the time has to nothing to do with predicting the action so we trained and tested with 3 features with 1 column as label and results came as shown below:

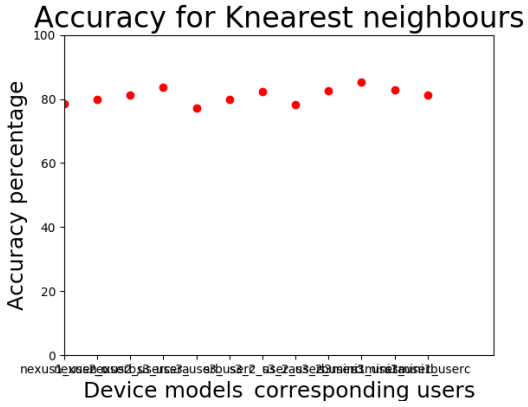

Figure 5.1: K nearest neighbour accuracy among various users and devices

5.3 Implementation Part:

Implementation part we completely done from the scratch and we used the parallel programming with 10 processorâĂŹs in order to solve this because this the algorithm

which is taking huge time when compared to any other above mentioned algorithms but the accuracy is pretty good when compared to other.

6 Logistic Regression

6.1 Description

This algorithm is usually taken to apply to a binary dependent variable.A logistic model is one where the probability of an event is a linear combination of independent or predictor variables. In this problem setting we are applying Logistic Regression Model in a multi class setting (using one vs all method) in order to predict what action the userâĂŹs is most probably going to take as recorded by a specific device by making use of the sensors.

6.2 Complexity of the Data

The data which we have to classify is having 3 input features to of each coordinate value and its label, by using the same parameters we trained the data (70 % of input) and tested with rest (30% of input) when we changed the parameters by considering the time also along with 3 coordinate parameters the results are falling from 83.33 % to 60 % so we came to know that the time parameters has not much importance because it makes sense because the time has to nothing to do with predicting the action so we trained and tested with 3 features with 1 column as label and results are shown in the figure below **??**

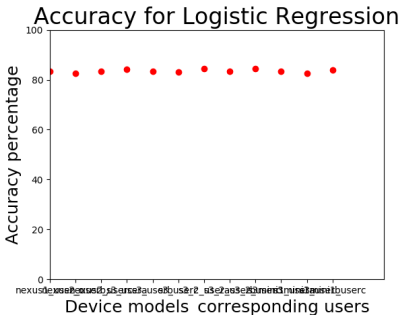

Figure 6.1: Logistic regression accuracy among various users and devices

6.3 Implementation Part

As discussed above in order to classify the Logistic regression model in the multivariate settings we are required to make use of one vs all method . So as there are 6 different actions a user can perform like Biking, Sitting, Standing, Walking, Stair Up and Stair down. By training the model on one vs all setting basically so considering the whole problem setting to be like two binary classes 1st one of the classification label considered as a class and 2nd all the other labels is considered another class.

6.4 References

http://mlwiki.org/index.php/One-vs-All_Classification

7 XGBoost

7.1 Description:

The reason why we used this is XGBoost algorithm because its a library designed and optimized for boosted tree algorithms which is very useful and opt to deal with current huge data we are having. As expected itâĂŹs giving the results faster than any other above mentioned algorithms.

7.2 Complexity of the Data:

The data which we have to classify is having 3 input features to of each coordinate value and its label, by using the same parameters we trained the data (70% of input) and tested with rest (30% of input) when we changed the parameters by considering the time also along with 3 coordinate parameters the results are falling from 85% to 62% so we came to know that the time parameters has not much importance because it makes sense because the time has to nothing to do with predicting the action so we trained and tested with 3 features with 1 column as label and results came as shown below 7.1

7.3 Implementation:

We didnt implemented this algorithm from scratch. we used the inbuilt libraries to trained and tested with above mentioned data and as mentioned above the execution is really fast and accuracy is mostly near to any other above mentioned algorithms.

7.4 References:

https://github.com/rasbt/mlxtend/issues/170

8 Random Forest

8.1 Description

Random forests or random decision forests are the method for classification, regression and other task that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

8.2 Complexity of the Data

The data which we have to classify is having 3 input features to of each coordinate value and its label, by using the same parameters we trained the data (70 % of input) and

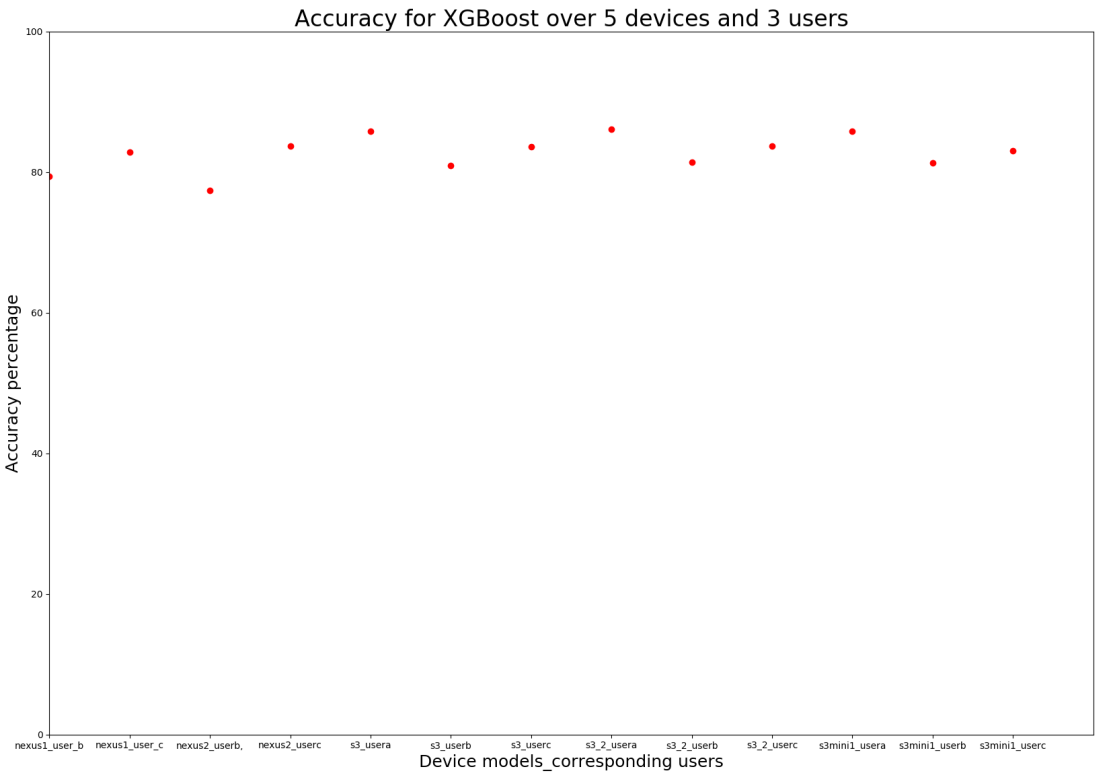

Figure 7.1: XG Boost accuracy among various users and devices

tested with rest (30 % of input) when we changed the parameters by considering the time also along with 3 coordinate parameters the results are falling from 83.33 % to 60 % so we came to know that the time parameters has not much importance because it makes sense because the time has to nothing to do with predicting the action so we trained and tested with 3 features with 1 column as label and results are shown in the figure below 8.1

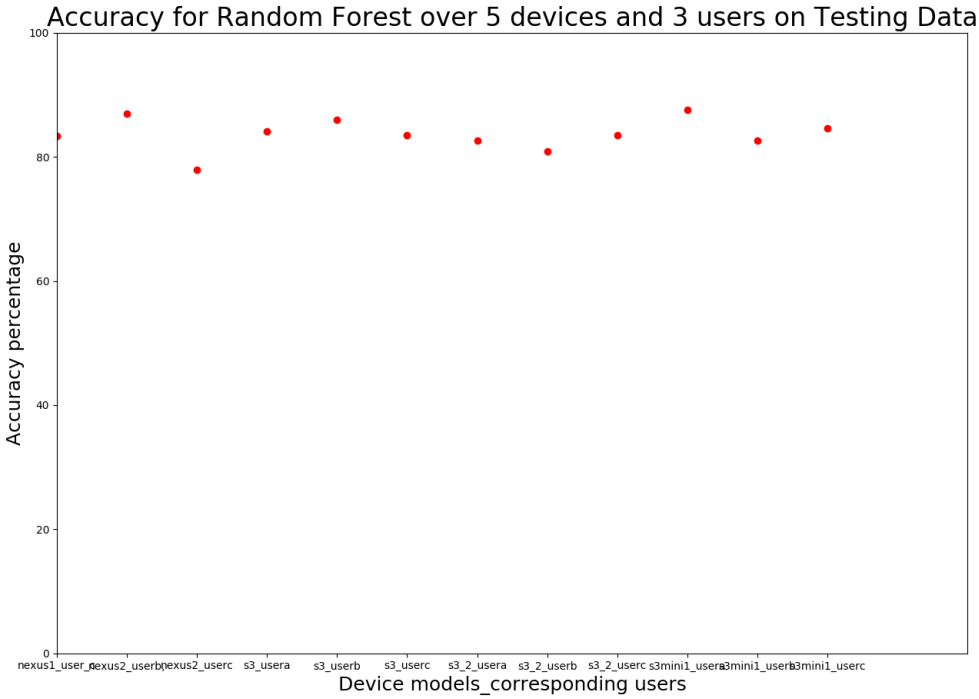

Figure 8.1: Random Forest accuracy among various users and devices

8.3 Implementation

We didn't implemented this algorithm from scratch. we used the inbuilt libraries to trained and tested with above mentioned data and as mentioned above the execution is really fast and accuracy is mostly near to any other above mentioned algorithms.

8.4 References

http://www.blopig.com/blog/2017/07/using-random-forests-in-python-with-scikit-learn/

9 Accuracy's

Device	User Name	K nearest neighbour	Logistic Regression	Random Forest	XG Boost
s31	a	83.7000	84.135	84.1354825779483	85.84
s32	a	82.2000	84.54	86.27748503324413	86.09
s3mini1	a	85.2000	83.31	87.53191253191254	85.85
nexus2	b	78.5000	82.5	79.78475139335035	77.43
s31	b	77.2000	83.333333	87.96765127008538	80.95
s32	b	78.3000	83.33	80.93207996822454	81.41
s3mini1	b	82.8000	82.604	82.60486476383734	81.31
nexus2	c	81.1000	83.319	84.1354825779483	83.69
s31	c	80.0000	83.13	79.96765127008538	83.59
s32	c	82.5000	84.458	83.45815501664289	83.73
s3mini1	c	81.2000	84	84.61466575291259	83.08

10 Conclusion

S No	Algorithms	Avg Accuracy	Avg Time
1	K nearest neighbour	81.154	966.315085
2	Logistic regression	83.5	312.5
3	XGBoost	86.04	7.134
4	Random Forest	82.99	8.191573

11 Acknowledgment

We have taken efforts in this project. However, it would not have been possible without the kind support and help of you Professor.Cho-Jui Hsieh. we would like to extend my sincere thanks to all of our team mates where everyone equally splited the work and completed the work Successfully.