A PROJECT REPORT

on

"HUMAN ACTIVITY RECOGNITION USING SMARTPHONE"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND COMMUNICATION ENGINEERING

BY

ABHINAV KUMAR	**1629119**
SAUBHAGYA ASHISH	**1629171**
SHUBHANGI GUPTA	**1629176**
SRIPRIYA SRIVASTAVA	**1629183**
SHEFALI PANDEY	**1629191**

UNDER THE GUIDANCE OF
PROF. RAJDEEP Chatterjee

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

April 2019

A PROJECT REPORT

on

"HUMAN ACTIVITY RECOGNITION USING SMARTPHONE"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER SCIENCE AND COMMUNICATION ENGINEERING

BY

ABHINAV KUMAR	**1629119**
SAUBHAGYA ASHISH	**1629171**
SHUBHANGI GUPTA	**1629176**
SRIPRIYA SRIVASTAVA	**1629183**
SHEFALI PANDEY	**1629191**

UNDER THE GUIDANCE OF
PROF. RAJDEEP CHATTERJEE

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

April 2019

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"HUMAN ACTIVITY USING SMARTPHONE"

submitted by

ABHINAV KUMAR	**1629119**
SAUBHAGYA ASHISH	**1629171**
SHUBHANGI GUPTA	**1629176**
SRIPRIYA SRIVASTAVA	**1629183**
SHEFALI PANDEY	**1629191**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2018-2019, under our guidance.

Date: 05/07/2019

(Rajdeep Chatterjee)
Project Guide

Acknowledgement

We take this opportunity to express our deep gratitude to all those helping hands without whom this project would have not been what it is. We take immense pleasure to express our thankfulness to our mentor Prof. Rajdeep Chatterjee for his constant motivation, timely suggestions which made our sail smooth through the odds we faced in our project. I would like to thank our Dean Prof. Samresh Mishra for giving us this opportunity to enhance our skills by giving this project and for motivating us throughout the time and also thank the complete faculty of Computer Science and Engineering who have provided us the knowledge to successfully complete the project. Last but not the least we would like to thank our friends who not only provided their helping hands in our project as and when required but also for becoming the first user of this portal and providing us with all the feedback which helped us improve our portal.

ABHINAV KUMAR
SAUBHAGYA ASHISH
SHUBHANGI GUPTA
SRIPRIYA SRIVASTVA
SHEFALI PANDEY

ABSTRACT

Human Activity Recognition has a wide application in medical research and human survey system.The aims is to identify the actions carried out by a person given a set of observations of himself/herself and the surrounding environment. Recognition can be accomplished by exploiting the information retrieved from Smartphone. The system uses smartphone data of accelerometer, gyroscope. The training data sets and test data sets, which are basically the standard data, are fed to the used algorithms(SVM, k-NN, Logistic Regression, Random Forest). The idea is that once the subject's activity is recognized and known, an intelligent computer system can then offer assistance. Based on the algorithms used Logistic regression achieved accuracy of approx 96%.

Keywords: Accelerometer, Gyroscope sensor, Human activity recognition, k-NN, Support vector machine, Logistic Regression, Random Forest.

Contents

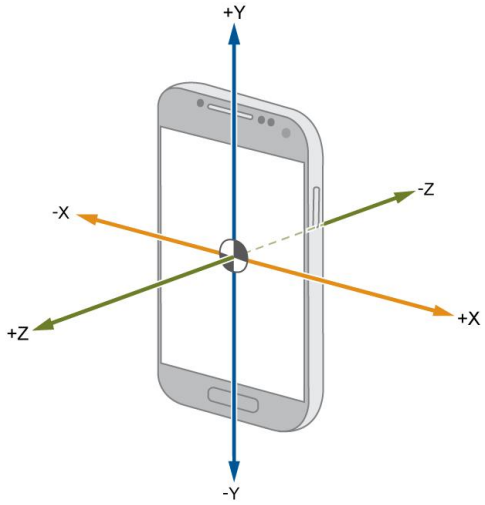
Chapter 1

Introduction

1.1. HUMAN ACTIVITY RECOGNITION

HAR has enabled various applications in different areas, such as, health care, security and entertainment. Initially,there were wearable motion sensors were used to recognize different physical activities. But now, there has been a shift towards mobile phones in recent years, because of the availability of various sensors which are already installed. Examples of such sensors are GPS, accelerometer, gyroscope and some other sensors.

Figure 1.1: Mobile Sensor



1.2. MACHINE LEARNING METHODS

In this section, four machine learning technique used in comparison are defined.

1.2.1. Logistic Regression

Logistic regression is a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Logistic regression is a classification algorithm, it is used to discriminate value like binary values(0,1) or true or false. It predicts the probability of occurance of an event by filling the data to a sigmoid function.

$$g(z) = \frac{1}{1+e^{-z}}$$

1.2.2. **k-NN**

In K-Nearest Neighbour classification, the output is a class membership. An object is classified by a plurality vote of its K-neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the case is simply assigned to the class of its nearest neighbor.

1.2.3. SUPPORT VECTOR MACHINE

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

1.2.4. **RANDOM FOREST**

Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.

1.3. MOBILE SENSORS

1.3.1. ACCELEROMETER

An accelerometer is a device that measures proper acceleration. Proper acceleration, being the acceleration of a body in its own instantaneous rest frame, is not the same as coordinate acceleration, being the acceleration in a fixed coordinate system. In short it an instrument for measuring the acceleration of a moving or vibrating body.

1.3.2. GYROSCOPE

Gyro sensors, also known as angular rate sensors or angular velocity sensors, are devices that sense angular velocity. Angular velocity. In simple terms, angular velocity is the change in rotational angle per unit of time. Angular velocity is generally expressed in deg/s (degrees per second).The gyroscope, adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist.

Chapter 2

Literature Survey

Here we first collected the data sets from kaggle.com on different human activities such as sitting, walking,walking upstairs, walking downstairs, laying, standing.The datasets are recorded from accelerometer and gyroscope of a smartphone. After that the data is fetch and fed to different classifiers such as SVM, Random Forest, Logistic Regression, and k-NN. After that accuracy was measured . Where, logistic Regression was found to be having the highest accuracy of approx 96.19%, Svm was having approx of 94.02% , Random Forest was having approx of 90.73%, and k-NN was having 90.02%. Also UI is made using ext js in which the front-end is having nine columns of data of body accelerometer and body gyroscope.The back-end is connected through sql table in which data is stored, which is further fetch in the UI.

Chapter 3

Software Requirements Specification

3.1. Anaconda

Anaconda is an open source free platform for Python and R programming languages for scientific use i.e., applications for machine learning,data science,predictive analytic,etc.It has been providing end-to-end projects experience.It is equipped with more than 1,500 packages along with virtual environment.In the project,anaconda is used for the package installation which provides greater dimensions in the data science field.Anaconda basically acts as an environment manager.For the updating purposes of its own self,Anaconda has it's own package manager known as "Conda" which happens to be a cross platform package.Anaconda install packages with dependencies being one of its advantages.It has several third party libraries such as pandas,numpy and scikit-learn.

3.2. Jupyter Notebook

Jupyter Notebook,being one of the tools of Anaconda, contains line codes and various visualizations which is a major breakthrough in data science field.It acts as a working and a living online notebook.Providing computational information,this has been used in this project.It is a free source web applications.It contains of explanatory text which eases the visualizations part.The use of Jupyter Notebook is numerical simulation,machine learning,data cleaning and transformation has been significantly noticed recently.It makes the visualization easy through graphical approach.It is an efficient way to iterate on python or machine learning for data analysis.It provides an interactive way of running codes on REPL Model I.e., Read-Eval-Print-Loop.Storing our code followed by keeping the markdown notes is what Jupyter Notebook provide.It is used for rapid prototyping which is one of the main features of this web-based interface.
Conclusively,Jupyter Notebook has been used in this project as a digital notebook which has provided visualizations through executing commands,taking notes and taking graphs etc.

3.3. Eclipse

Eclipse is a platform which is an integrated development environment used in computer languages specifically in Java SE.It was developed by Eclipse foundation.It uses several programming languages such as Java programming language,C/C++,Python,Ruby,Pearl etc. Integrated development environment,rich client applications,and other tools are developed using Eclipse platform.It is a free to download and install which allows Eclipse to be modified and distributed.In a normal text editor the lack of deep structural understanding of the codes makes it less reliable .Henceforth,making Eclipse IDE more reliable.The format understands the code which eases its dependencies on other projects.All of this enables productivity tools which makes it a great platform for writing codes.It has feature which offer tools such as code formatting,auto completion,type error reporting etc.

Chapter 4

Requirement Analysis

Anaconda equipped with more than 1,500 packages along with virtual environment.In the project,the platform is used for the coding in Machine Learning which enables to use different packages and libraries in the coding part.

It provides package installation with dependencies.It has several third party libraries which are used such as pandas,numpy and scikit-learn under Matplot library with extension .lib.

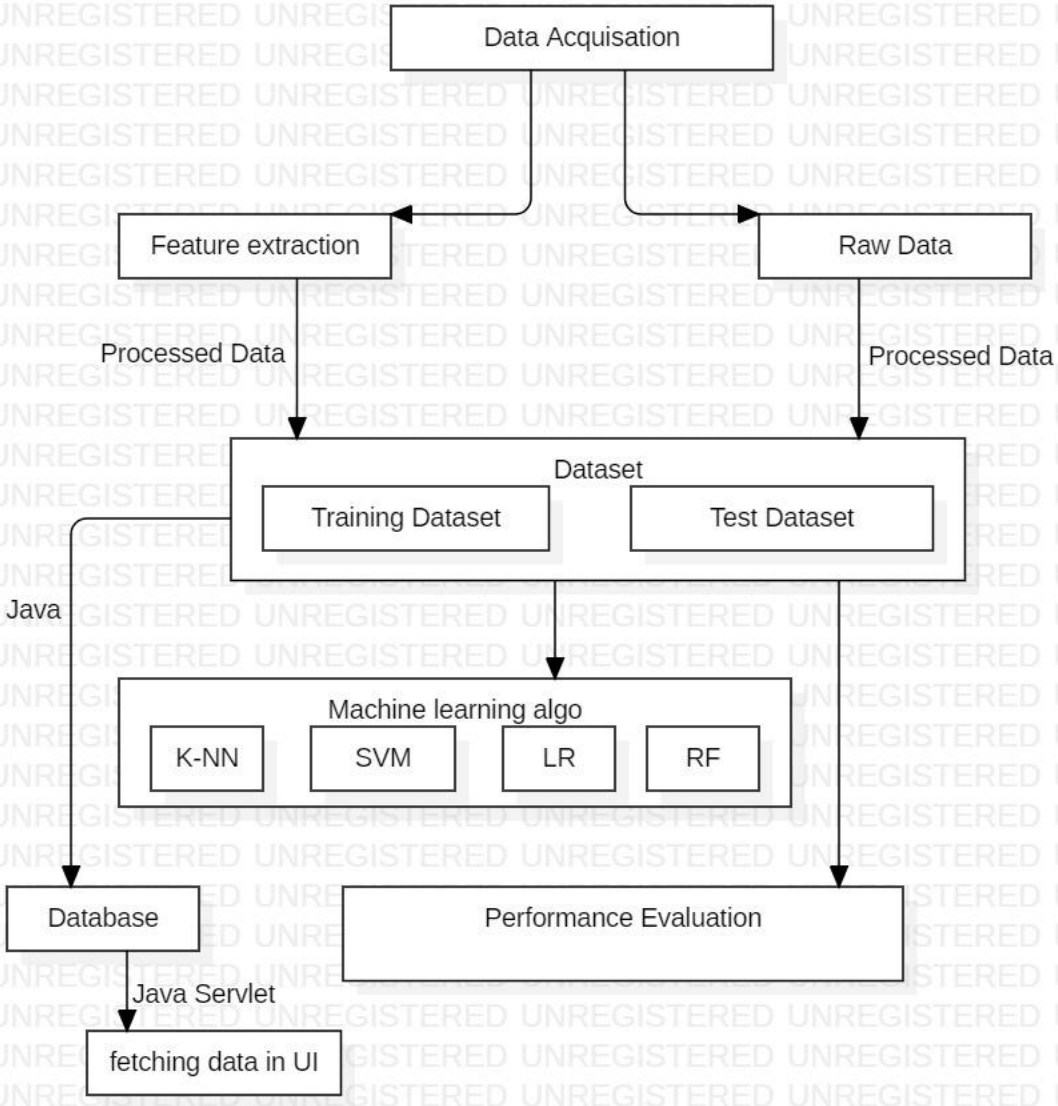
Jupyter Notebook being one of the tools of Ipython and anaconda specifically,stores editable document.The Notebook is used for front end and back end in which apart from running the code it is used to store code and output along with markdown notes.The lines of codes are run one at a time and then can be edited accordingly.

Eclipse is an integrated development environment which is used in this project for deep structural understanding of the codes.It is also used for writing codes for front end and back end.The codes are modifies accordingly.Codes are run for type error checking and code formatting.

Chapter 5

System Design

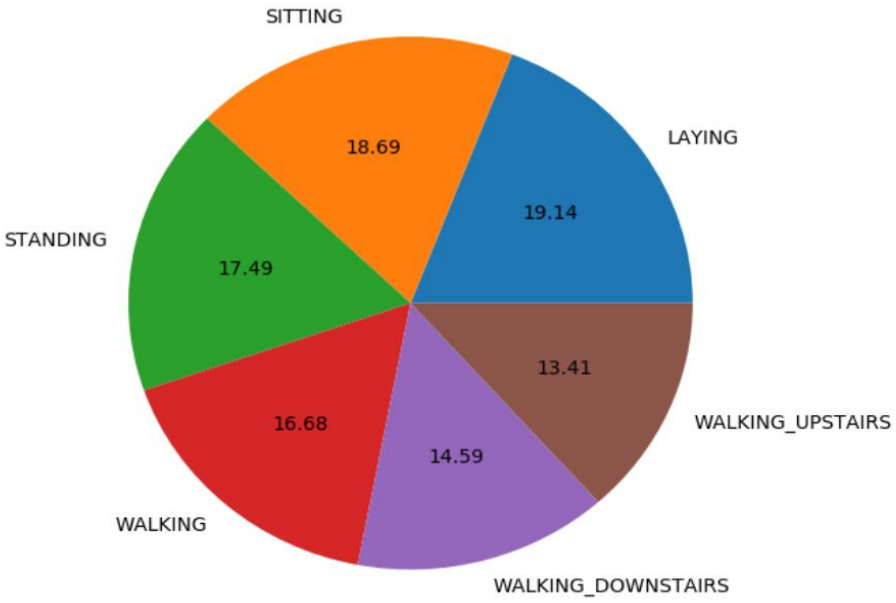
5.1 WORK FLOW CHART



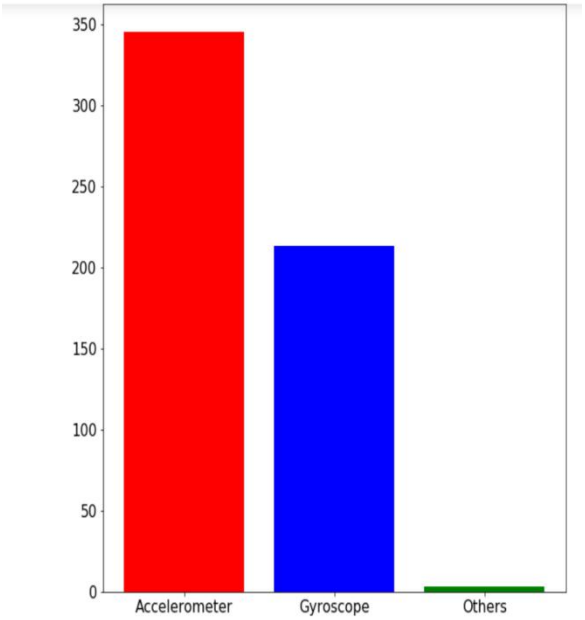
Chapter 6

System Testing

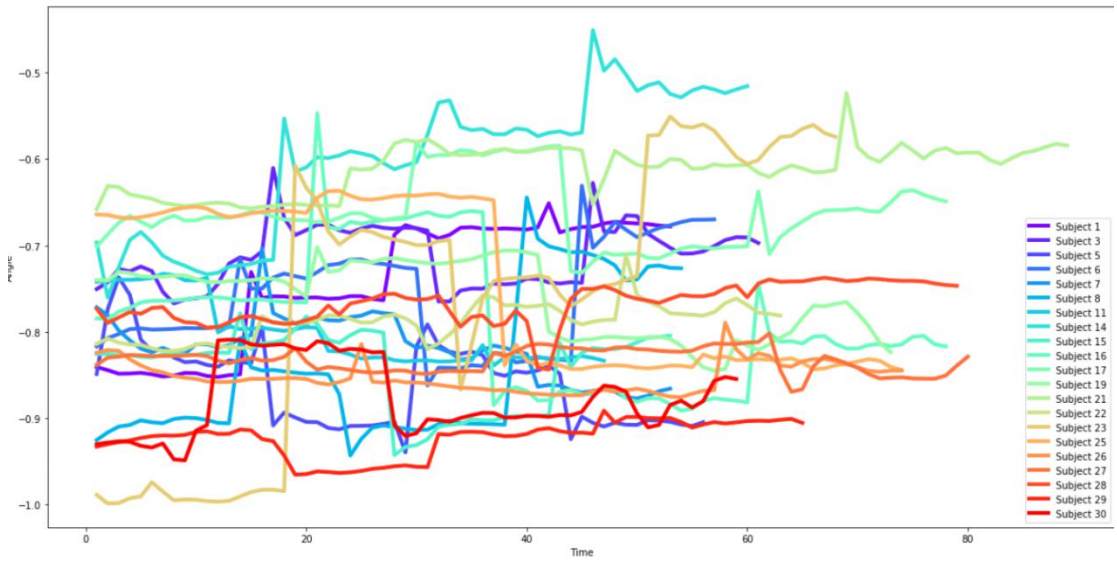
This basically comprises of the accuracy gained by the input(x) and output(y) data provided in the dataset.



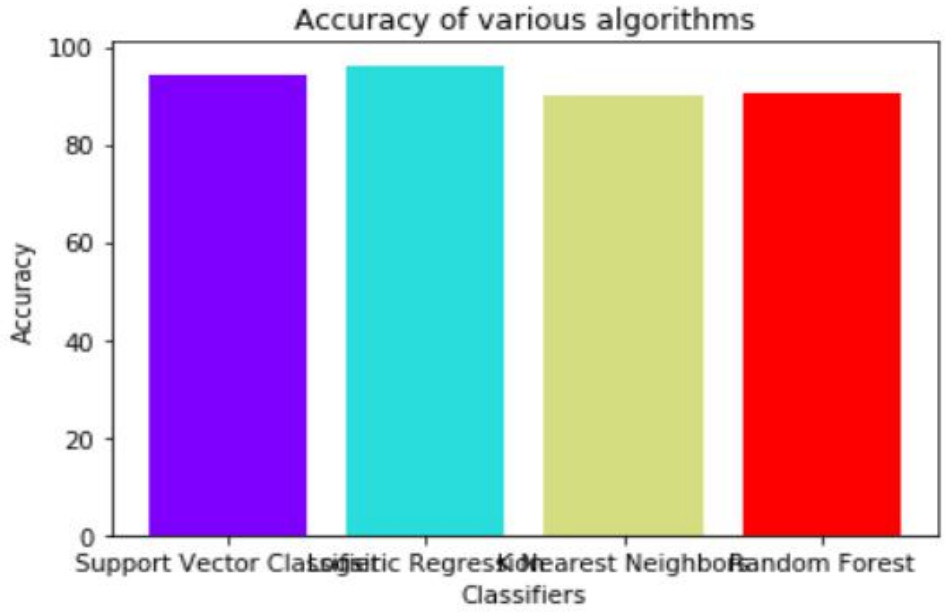
This pie chart comprises of counting the number of records for each of the activity specified in the chart in a sorted manner.



This bar chart comprises of the values of the three most important variables in the dataset i.e. Accelerometer, Gyroscope and the other random variables such as Gravity, entropy etc.



This is an overlapping chart showing the activity of arbitrary 30 subjects together. This chart is showing an angle between the input(X) values and the mean gravity v/s time for the subjects given in the dataset at a random.



This bar chart is showing the accuracy of 4 different algorithms used in the machine learning coding.
With Logistic Regression showing the highest accuracy of 96%, the different algorithms used here are Support Vector Classifier, Logistic Regression, K Nearest Neighbors Classifier and the Random Forest Classifier.

Chapter 7

Implementation

Firstly, we developed the code to load the dataset and evaluate some machine learning algorithms. The goal was to achieve the maximum accuracy on the test dataset.

The results of methods using the dataset provides a baseline for any methods developed for the raw data version.

The section is divided into five parts :-

- Load Dataset
- Define Models
- Evaluate Models
- Summarize Results

7.1 LOAD DATASET

The first step is to load the train and test input (X) and output (y) data.The input data is in CSV format where the columns are separated via the whitespace.

```
training_data = pd.read_csv(r'D:\Human recognition using smartphone\train.csv')
testing_data = pd.read_csv(r'D:\Human recognition using smartphone\test.csv')

# For training data
print("Training Data: {}".format(training_data.shape))
print("Null values present in training data: {}".format(training_data.isnull().values.any()))

# For testing data
print("Testing Data: {}".format(testing_data.shape))
print("Null values present in testing data: {}".format(testing_data.isnull().values.any()))
```

7.2. DEFINE MODELS

Here we define the list of machine learning models to evaluate on the dataset.

This evaluation is done by a set of non linear and ensemble algorithms specifically :

- K- Nearest Neighbors

```
# K Nearest Neighbors
clf = KNeighborsClassifier().fit(X_train, y_train)
prediction = clf.predict(X_test)
accuracy_scores[2] = accuracy_score(y_test, prediction)*100
print('K Nearest Neighbors Classifier accuracy: {}%'.format(accuracy_scores[2]))
```

```
K Nearest Neighbors Classifier accuracy: 90.02375296912113%
```

- Support Vector Machine

```
# Support Vector Classifier
clf = SVC().fit(X_train, y_train)
prediction = clf.predict(X_test)
accuracy_scores[0] = accuracy_score(y_test, prediction)*100
print('Support Vector Classifier accuracy: {}%'.format(accuracy_scores[0]))
```

```
Support Vector Classifier accuracy: 95.24940617577197%
```

- Logical Regression

```
# Logistic Regression
clf = LogisticRegression().fit(X_train, y_train)
prediction = clf.predict(X_test)
accuracy_scores[1] = accuracy_score(y_test, prediction)*100
print('Logistic Regression accuracy: {}%'.format(accuracy_scores[1]))
```

```
Logistic Regression accuracy: 96.13165931455717%
```

- Random Forest

```
# Random Forest
clf = RandomForestClassifier().fit(X_train, y_train)
prediction = clf.predict(X_test)
accuracy_scores[3] = accuracy_score(y_test, prediction)*100
print('Random Forest Classifier accuracy: {}%'.format(accuracy_scores[3]))
## Output:
# Random Forest Classifier accuracy: 89.68442483881914%
```

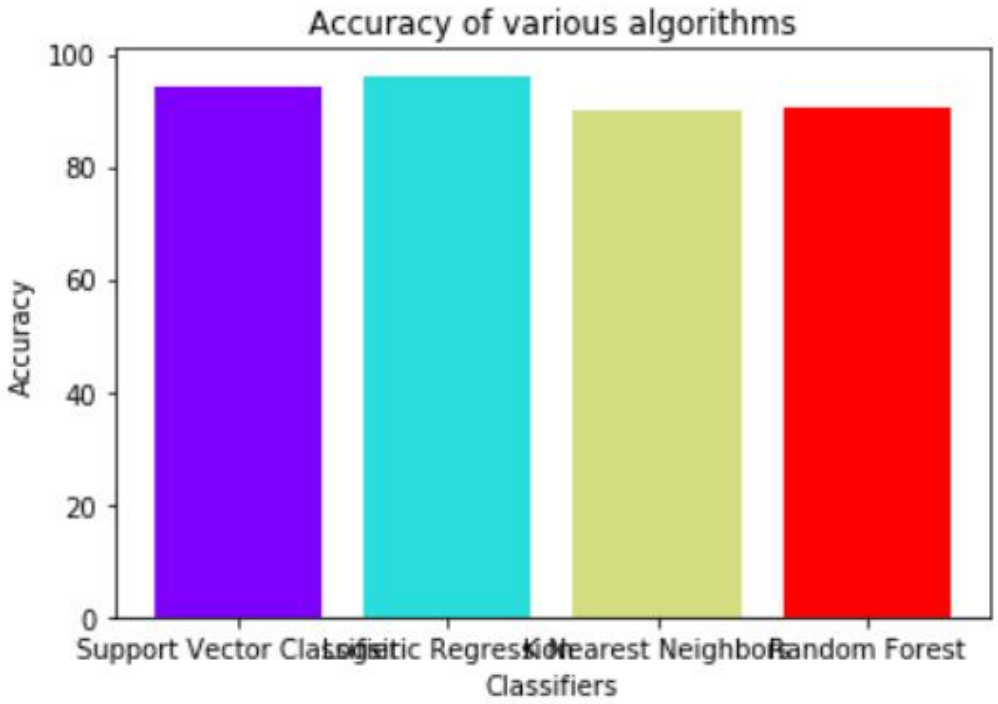
```
Random Forest Classifier accuracy: 90.60061079063453%
```

EVALUATE MODELS

This evaluation basically shows that the performance of each model in terms of prediction and accuracy.

SUMMARISE RESULTS

We sort all the results by the models in the classification accuracy in terms of bar charts.

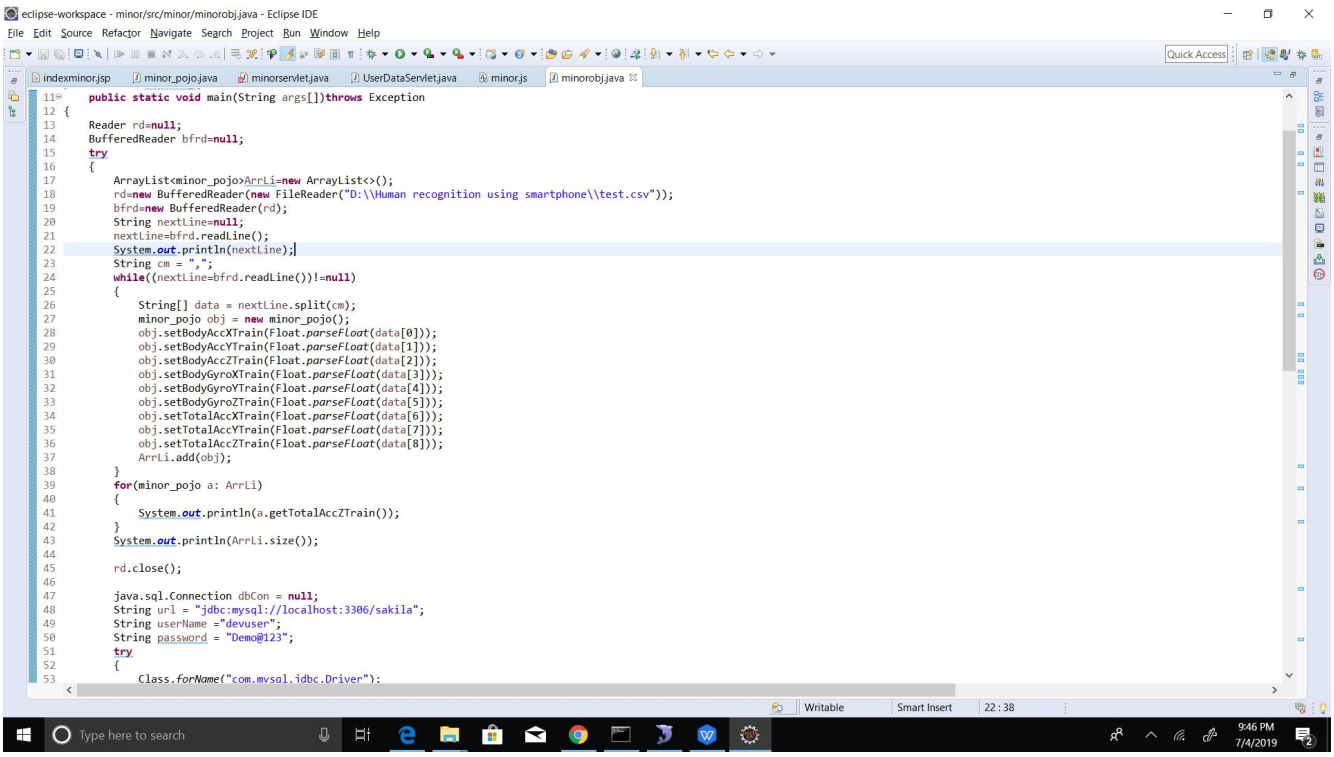


Chapter 9

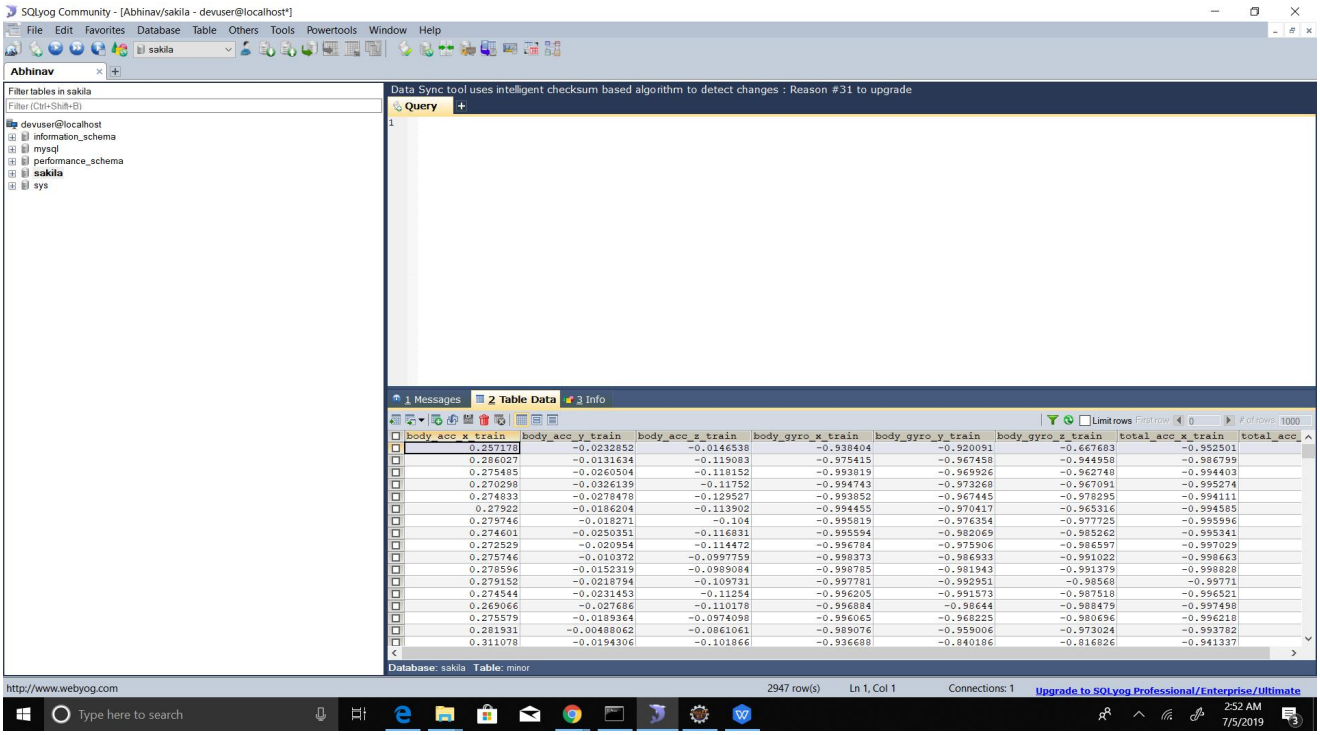
Screenshots of Project

9.1 Back-end

Storing data of the dataset into database through java--

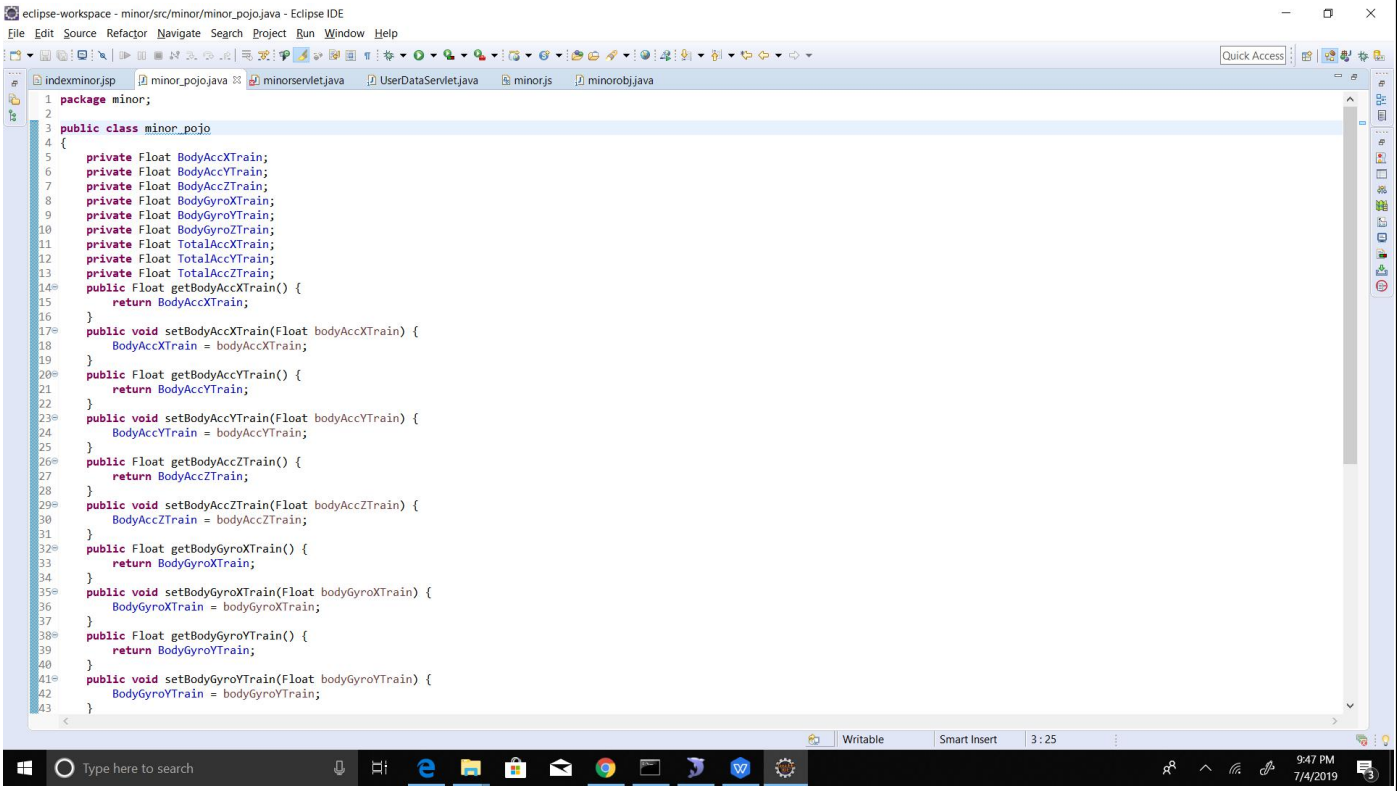


Data Stored in the database--

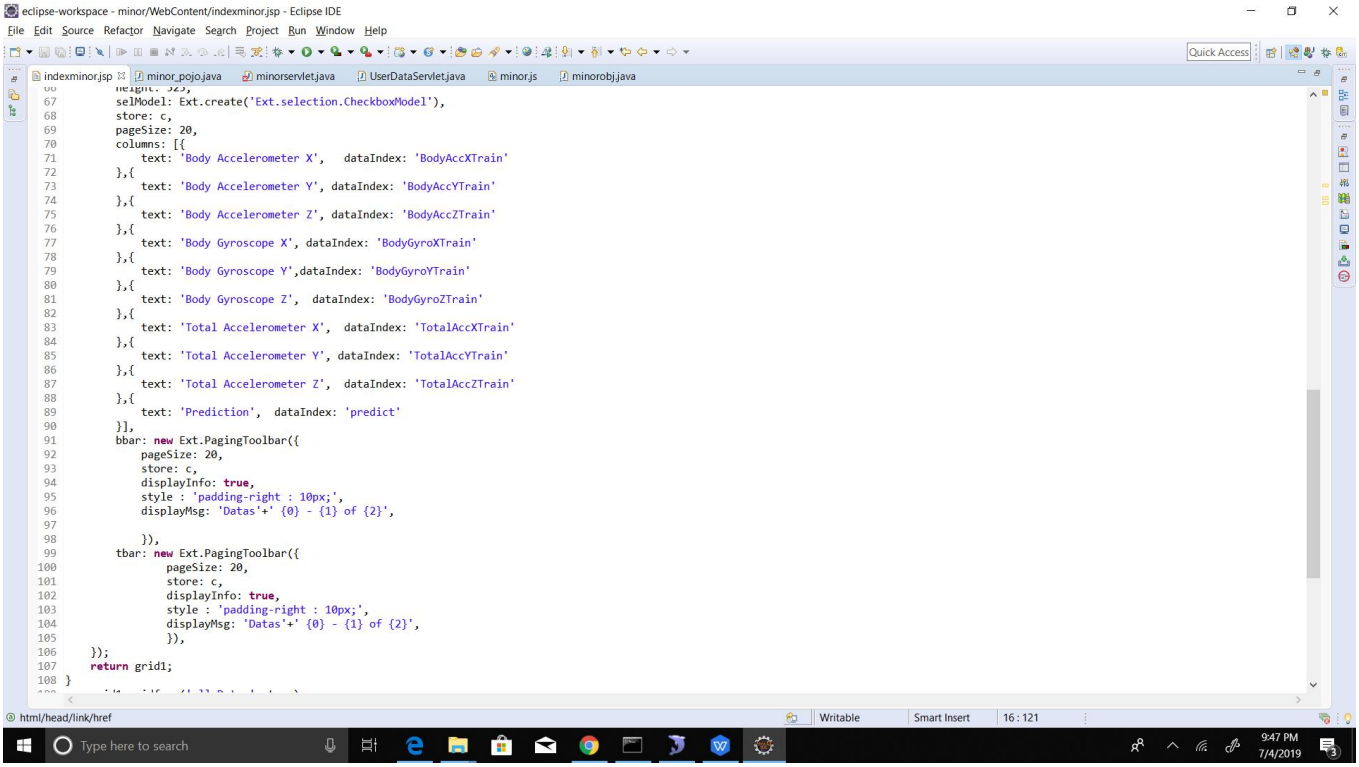


9.2 Front-end

Fetching Data from database to UI--



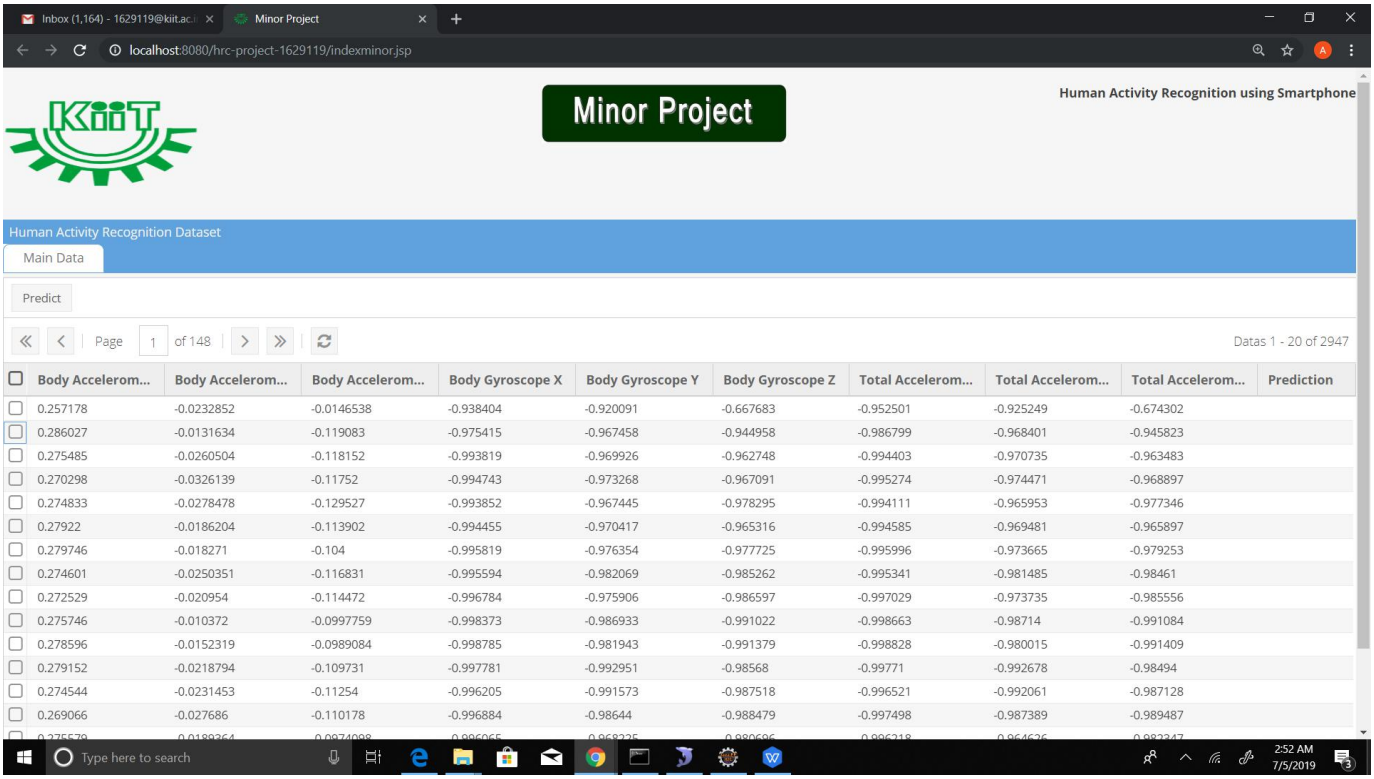
UI Grid Creation-



```
        selModel: Ext.create('Ext.selection.CheckboxModel'),
        store: c,
        pageSize: 20,
        columns: [{
            text: 'Body Accelerometer X',   dataIndex: 'BodyAccXTrain'
        },{
            text: 'Body Accelerometer Y', dataIndex: 'BodyAccYTrain'
        },{
            text: 'Body Accelerometer Z', dataIndex: 'BodyAccZTrain'
        },{
            text: 'Body Gyroscope X', dataIndex: 'BodyGyroXTrain'
        },{
            text: 'Body Gyroscope Y',dataIndex: 'BodyGyroYTrain'
        },{
            text: 'Body Gyroscope Z',  dataIndex: 'BodyGyroZTrain'
        },{
            text: 'Total Accelerometer X',  dataIndex: 'TotalAccXTrain'
        },{
            text: 'Total Accelerometer Y', dataIndex: 'TotalAccYTrain'
        },{
            text: 'Total Accelerometer Z',  dataIndex: 'TotalAccZTrain'
        },{
            text: 'Prediction',  dataIndex: 'predict'
        }],
        bbar: new Ext.PagingToolbar({
            pageSize: 20,
            store: c,
            displayInfo: true,
            style : 'padding-right : 10px;',
            displayMsg: 'Datas'+' {0} - {1} of {2}',

            }),
        tbar: new Ext.PagingToolbar({
                pageSize: 20,
                store: c,
                displayInfo: true,
                style : 'padding-right : 10px;',
                displayMsg: 'Datas'+' {0} - {1} of {2}',
                }),
    });
    return grid1;
}
```

Output of UI--



Body Accelerom...	Body Accelerom...	Body Accelerom...	Body Gyroscope X	Body Gyroscope Y	Body Gyroscope Z	Total Accelerom...	Total Accelerom...	Total Accelerom...	Prediction
0.257178	-0.0232852	-0.0146538	-0.938404	-0.920091	-0.667683	-0.952501	-0.925249	-0.674302	
0.286027	-0.0131634	-0.119083	-0.975415	-0.967458	-0.944958	-0.986799	-0.968401	-0.945823	
0.275485	-0.0260504	-0.118152	-0.993819	-0.969926	-0.962748	-0.994403	-0.970735	-0.963483	
0.270298	-0.0326139	-0.11752	-0.994743	-0.973268	-0.967091	-0.995274	-0.974471	-0.968897	
0.274833	-0.0278478	-0.129527	-0.993852	-0.967445	-0.978295	-0.994111	-0.965953	-0.977346	
0.27922	-0.0186204	-0.113902	-0.994455	-0.970417	-0.965316	-0.994585	-0.969481	-0.965897	
0.279746	-0.018271	-0.104	-0.995819	-0.976354	-0.977725	-0.995996	-0.973665	-0.979253	
0.274601	-0.0250351	-0.116831	-0.995594	-0.982069	-0.985262	-0.995341	-0.981485	-0.98461	
0.272529	-0.020954	-0.114472	-0.996784	-0.975906	-0.986597	-0.997029	-0.973735	-0.985556	
0.275746	-0.010372	-0.0997759	-0.998373	-0.986933	-0.991022	-0.998663	-0.98714	-0.991084	
0.278596	-0.0152319	-0.0989084	-0.998785	-0.981943	-0.991379	-0.998828	-0.980015	-0.991409	
0.279152	-0.0218794	-0.109731	-0.997781	-0.992951	-0.98568	-0.99771	-0.992678	-0.98494	
0.274544	-0.0231453	-0.11254	-0.996205	-0.991573	-0.987518	-0.996521	-0.992061	-0.987128	
0.269066	-0.027686	-0.110178	-0.996884	-0.98644	-0.988479	-0.997498	-0.987389	-0.989487	

Chapter 10

Conclusion and Future Scope

10.1 Conclusion

Need of understanding human activities has intrigued demands in the new era technology.It has broad applications in various fields such as medical arena or survey system which needs to be mentioned thoroughly.

In this project,a smartphone-based recognition system is designed which provides assistance in monitoring various human activities such as :-

- Standing
- Walking
- Waling upstairs
- Walking downstairs
- Sitting
- Laying

It is designed to assist the health-care domain to a great extent.The concept of smartphones as low costs sensor to identify human activities is used in this project.The use of gyroscope and accelerometer has helped in activity recognition.The technology's urgent need and requirements of regularly monitoring human activities adds another dimension to the need of human-activity recognition.

Conclusively,logistic regression is the optimal solution to our problem.It gives the best performance and is promising to run on smartphones.

10.2 Future Scope

Human activity recognition tends to be an important aspect in the AI and ML field.Numerous facts,deliberate efforts have been put up in establishing the very important assets of this very recognition.In computer mission,it can act as a basis for many applications including health-care surveillance and human-computer relationship.Various technologies have been developed and it is still updating.Still it faces various challenges when handling realistic approaches.It has to be tackled deliberately in order to achieve the maximum potential.The use of Human Activity Recognition using smartphones can add up to several vital applications which can tremendously benefit the future scenario.

References

[1] *HUMAN ACTIVITY RECOGNITION USING SMARTPHONE*; BY-Erhan Bulbul ; Aydin Cetin ; Ibrahim Alper Dogru

[2] http://machinelearningmastery.com

[3] http://www.kaggle.com/uciml/human-activity-recognition-with-smartphones

[4] https://towardsdatascience.com/human-activity

[5] http://www.geeksforgeeks.org

[6] http://www.wikipedia.org