CIS.735.M001.SPRING23.Machine Learning for Security.

Project - II

Master of Science

In

Computer & Information Sciences

Submitted by

Trinath Reddy Inturi (476506643), Deepthi Konda (288773035) & Meghana Pusapati (739822372)

Under the Guidance of

Prof., Vir Phoha

Department of Electrical Engineering and Computer Science

SYRACUSE UNIVERSITY

Problem Definition:

The objective is to develop a model that can automatically recognize and predict activities like standing, walking, moving forward and moving backward from the time series data generated by accelerometer and gyroscope using HTC smartphone of 17 users. The goal is to provide an accurate and reliable tool for activity recognition and prediction that can be used in various applications such as healthcare, sports, and fitness tracking.

Problem Solution:

Given the time series data of accelerometer, gyroscope, and linear accelerometer, we first read them into a data frame with columns as timestamp, x, y, z respectively. We use the accelerometer data to find out if the user is standing or moving by calculating the magnitude of the users' readings and the **Mean** of all the magnitude values. We are considering the threshold as the mean value and the step length as 0.7 meters which is 2.2 feet (In general). Using this we had planned to perform the activity recognition on the given data.

We used the accelerometer data and the gyroscope data to predict the movement or to know, if the user is walking, ascending, or descending. For this we are considering a threshold value by taking the mean of the magnitude calculated.

We then use the models Logistic Regression and Random Forest to train the data which contains the labels of the activities we obtained and then predicted the activity labels for the given testing data.

Explanation:

Data Preprocessing:

Given the time series data of accelerometer, gyroscope, and linear accelerometer, we first read them into a data frame with columns as timestamp, x, y, z respectively. A new column named user is created and all the users are added.

```
data_dir = 'data/Training Data'
users = ['User001', 'User002', 'User003', 'User004', 'User005', 'User006', 'User007',
```

Feature Engineering:

We then calculate the magnitude of the accelerometer data in three dimensions (x, y, and z) using the Pythagorean theorem. We calculate the square root of the sum of squares of each axis of the data and assign the resulting values to the magnitude column created in the data frame. After this the mean is calculated to set a threshold value and the step length is set to 0.7 meters (2.2 Feet) which is the average step length of human beings. A mean magnitude value is calculated using the rolling window.

Then we created two columns, “peak” and “step”. The "peak" column is a Boolean column that indicates whether the value in the mean magnitude column is greater than a predefined threshold value. The "step" column contains True for each row where the peak value is different from the peak value of the previous row (using the shift() function), and where the peak value of the current row is True. Essentially, this identifies the start of a new step since a step starts when the mean magnitude value crosses the threshold value and then goes back down below it.

So, peak is used to identify when the magnitude of the acceleration crosses the threshold value, while step is used to identify when a new step starts. Finally, the number of steps is calculated by adding all the values of step and distance is calculated by multiplying step length with number of steps.

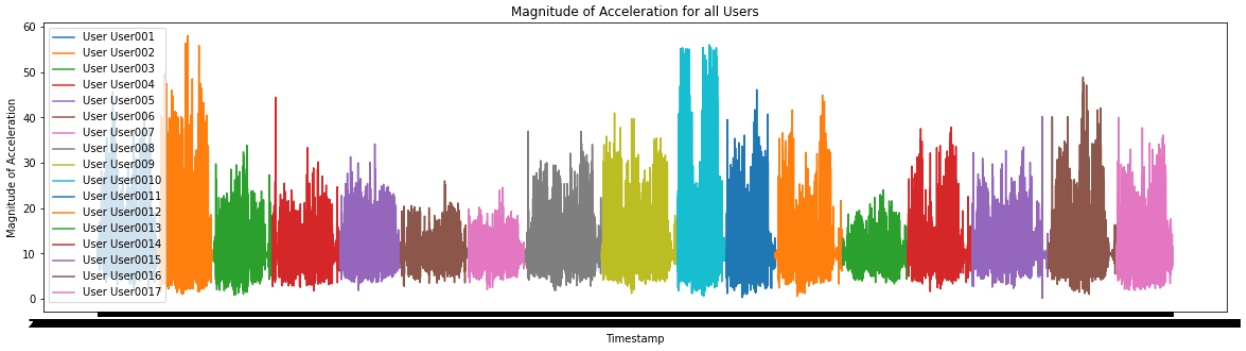
```
In [45]: accel_main.groupby(by="activity")["magnitude"].count()

Out[45]: activity
         Moving       1353
         Standing    96072
         Name: magnitude, dtype: int64

In [46]: gyro_main.groupby(by="activity")["magnitude"].count()

Out[46]: activity
         Ascending     31212
         Descending    30471
         Standing      35058
         Walking         688
         Name: magnitude, dtype: int64

In [47]: accel_main.groupby(by="user")["step"].sum()

Out[47]: user
         User001      67
         User0010     11
         User0011     36
         User0012     96
         User0013    104
         User0014     56
         User0015     64
         User0016     72
         User0017     60
         User002      35
         User003      87
         User004     108
         User005     115
         User006     164
         User007     128
         User008      73
         User009      77
         Name: step, dtype: int64
```

We define a function called gyro_activity() which takes in the gyroscope data and the threshold value which we declared taking the mean as two parameters. The function initializes an empty list called "gyro_activity". Then, for each row in the "gyro_data" DataFrame, it extracts the value of the "mean_mag" column and assigns it to a variable called "mag".

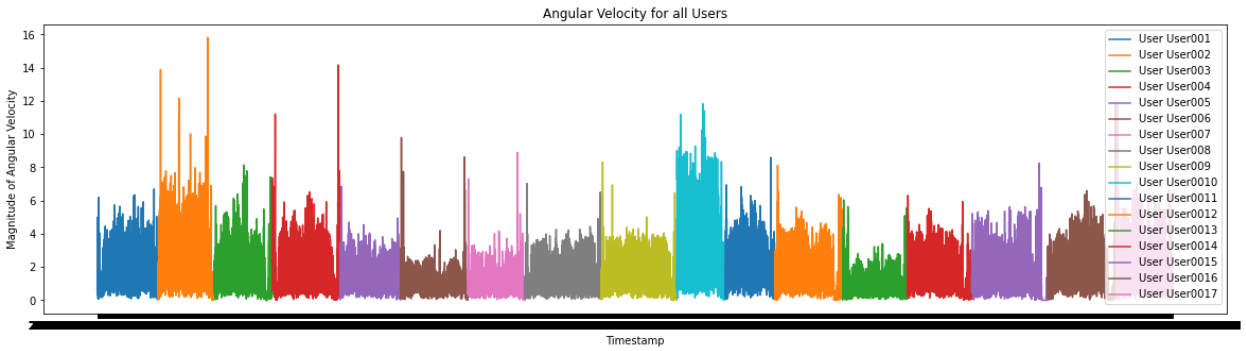
Plotting of Accelerometer and Gyroscope Magnitude values for Observations:

We then determine the activity type based on “mag” and the value of z which is the angular velocity that helps to describe the movement of the user. If "z" is positive and "mag" is greater than the threshold, the activity type is "Ascending". If "z" is negative and "mag" is greater than the threshold, the activity type is "Descending". If "mag" is greater than or equal to the threshold, the activity type is "Walking". Otherwise, the activity type is "Standing". Finally, the function appends the activity type to the "gyro_activity" list and returns it. The returned list contains the activity type for each time step in the "gyro_data" Data Frame.

Graph of timestamp on x axis and magnitude of acceleration on y axis for accelerometer data of all the users:



Graph of timestamp on x axis and magnitude of angular velocity (Gyroscope) on y axis data of all the users:



Data Predictions and the comparison of results:

The next step is to perform the classification using logistic regression and Random Forest Classifier to predict activity labels for the testing data with the help of above generated results. Here are some observations below:

Count of each activity predicted using “groupby” for both accelerometer and gyroscope data.

Further code comments

2.2.3 Comparing Results

```
In [91]: print(test_accel_log.groupby(by="activity")["magnitude"].count())
         print(test_accel_rff.groupby(by="activity")["magnitude"].count())
```

```
activity
Moving         82
Standing    95463
Name: magnitude, dtype: int64
activity
Moving         82
Standing    95463
Name: magnitude, dtype: int64
```

2.2.6 Comparing Results

```
In [97]: print(test_gyro_log.groupby(by="activity")["magnitude"].count())
         print(test_gyro_rff.groupby(by="activity")["magnitude"].count())
```

```
activity
Ascending     35934
Descending    43291
Standing      15313
Walking        1005
Name: magnitude, dtype: int64
activity
Ascending     35934
Descending    43291
Standing      15313
Walking        1005
Name: magnitude, dtype: int64
```