Assignment 2
CE 784A: Machine Learning and Data Analytics for Civil Engineering Applications

Submission Deadline: 6th March, 2020 11:59 PM IST

Total Marks: 40

Instructions: Use Jupyter notebook to work on this assignment using Python language. All cell outputs and visualizations should be visible and necessary comments should be put to make the code readable. Once the code is ready, save it as a pdf too. Create a zip file containing the pdf and the *.ipynb file* obtained from Jupyter Notebook. You should upload this zip file directly in the hello iitk assignment page. Please avoid sending your submission by email since they will not be graded.

In this assignment, the task will be to develop models for transportation mode detection classification. Transportation Mode Detection involves finding out the specific mode of transport (i.e., bus/car/train/walking/running etc.) the user is involved. Smartphone sensors have become an important data source for transportation mode detection. A sample paper details is given below to get further details on transportation mode detection using smartphones:

Hemminki, S., Nurmi, P., & Tarkoma, S. (2013). Accelerometer-based transportation mode detection on smartphones. *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems - SenSys '13*. 11th ACM Conference. https://doi.org/10.1145/2517351.2517367

Dataset:

The dataset in this Assignment homepage of Hello IITK.

The dataset consists of a zip folder named "cleaned.zip", which contains the csv file named "cleaned.csv". Unzip the folder to get the main csv data file.

The schema of the dataset is as follows: ***user, timestamp, x, y, z, class***

user: unique user id given to each participant
timestamp: timestamp of data recorded
x: Accelerometer reading X-axis obtained from sensor
y: Accelerometer reading Y-axis obtained from sensor
z: Accelerometer reading Z-axis obtained from sensor
class: labeled class i.e., transportation mode

There are 6 different transportation mode labels present in this dataset, namely bike, bus, car, e-bike, train, walk. The task of this assignment will be to build a classification model to label a sequence into one of these classes correctly. Note, each user can have multiple sequences as you can see in one of the sample plot from the data in the Figure 1, given below where the user shown in the row 1 of the figure had 4 distinct sequences: two of the label "car" and two of the label "walk". [I used a software Tableau for quick visualization of this dataset. We will also learn briefly about this tool later in our class.]

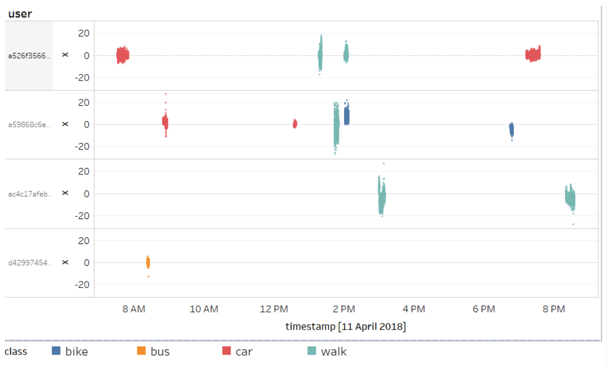

Figure 1. Sample plot from the dataset

The data is obtained from https://github.com/QROWD/QROWD_TMD. Refer to this github page if you want further details on the dataset. The detailed questions are provided next along with the marks allotted for each question. The specific outputs that need to be provided are mentioned and highlighted in italics. All figures should have proper axis labels, title, and legends details (if any). To better explain your code, you can output any other relevant details.

1. **Read the dataset "cleaned.csv"**. You can use python library such as pandas for reading the dataset. *a) Output the number of rows, columns, and columns names of the input data.* *b) Find out the number of unique users present in the dataset and output the value (Column "user")*	1+1
2. ***Determine the number of unique sequences*** of different transportation modes present in the dataset. For example, as stated before, the particular user shown in the row 1 of Figure 1 had 4 distinct sequences on 11th April 2018: two sequences of the class "car" and the remaining two of the class "walk". If there is any discontinuity of greater than 10 seconds, assume it to be a separate sequence i.e., the minimum time gap between two unique sequences is assumed to be 10 seconds. *a) Output using a table the number of unique sequences present for each transportation mode for each user.* *b) Output the time taken for your code to run this particular section of code (determining number of unique sequences).* You can use "timeit" function in Python to obtain the time taken to run a particular section of your code. Note, try to avoid "for loops" for determining the sequences since it can take a lot of time to process such a large file sequentially.	4+1
3. Time window partition: Divide the dataset into time windows so that features can be extracted for each time window. Time window partition is to be done for each sequence separately for each transportation mode and user. Time window partition is one of the many techniques used to generate features from continuous time series sequences. Figure 2 shows a demo sample plot of time window partition with 30 seconds window length (for demo purpose only). For this question, **keep the time *window length as 5 seconds*** (this is a hyper-parameter, which you can later try to change too). Remove sequences less than 5 seconds. *Output the number of sequences you have obtained for each transportation mode and time taken to run this particular section of code.*	5

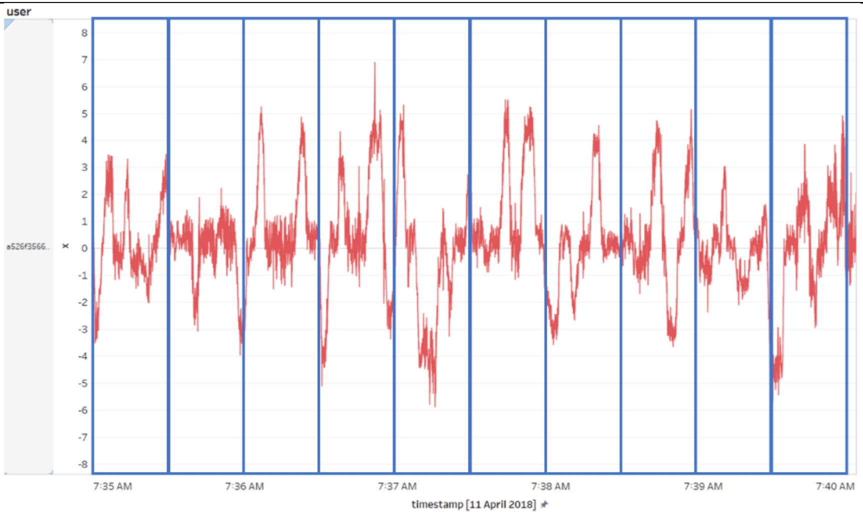

Figure 2. Time window partition with 30 sec length (Shown only for demo purpose)

4. ***Extract features for each time window partition***. Feature extraction is the main component of any machine learning model. For this assignment, extract features such as minimum, maximum, average, and standard deviation for each time window. Therefore, when you extract these 4 features for each of 3 acceleration variables (x, y, and z), you will have 12 features in total (x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}, …). *Use box and whisker plot for each feature and transportation mode.* Each figure will be for a particular feature and each figure will contain 6 box and whisker plots (side-by-side) for each transportation mode. Overall, there will be total 12 figures for 12 features. *Provide your interpretation of the visualization plots that you have obtained.*	8
5. **Create a balanced dataset**: The number of time window partitions obtained for each class can be different. Create a balanced dataset such that number of partitions (which will correspond to each data point) is same for each class. For example, say, Class A, Class B, and Class C has 25, 20, and 100 data points (time windows) respectively. Since minimum number of data points is 20 (for Class B), select 20 data points randomly from Class A and Class C and discard the remaining data points such that each class has 20 points, thereby creating a balanced data set. The numbers provided here are sample numbers and the actual numbers that you obtain for each transportation mode will be obviously different. *Output the number of data points obtained for each transportation mode in the balanced dataset that you prepared.*	3
6. **Training and test data split**: Split the balanced dataset obtained in the previous step into training, validation, and test data set using 60:20:20 split for train: validation: test. *Output the number of data points in training, validation, and test dataset.*	2
7. **Train different shallow supervised machine learning models** to obtain the best accuracy on the test data using your training and validation data. You need to try at least logistic regression, SVM, and ANN for this assignment. Emphasis should be to obtain the best possible test accuracy and therefore choose model architecture and/or model hyper-parameters accordingly. *Output the training and test accuracy of each ML model that you have applied. Also, output the details of the ML model that achieved best test accuracy based on your analysis.*	15

Reference Materials: The following resources might be helpful in the context of this assignment.

1. Ballı, S., & Sağbaş, E. A. (2018). Diagnosis of transportation modes on mobile phone using logistic regression classification. *IET Software*, 12(2), 142–151. https://doi.org/10.1049/iet-sen.2017.0035
2. Hemminki, S., Nurmi, P., & Tarkoma, S. (2013). Accelerometer-based transportation mode detection on smartphones. *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems - SenSys '13*. 11th ACM Conference.
3. http://cs.unibo.it/projects/us-tm2017/index.html and http://cs.unibo.it/projects/us-tm2017/tutorial.html
4. https://www.kaggle.com/alefsegura/kernel99234a943c