# Machine Learning Engineer Nanodegree
# Capstone Proposal

Anshul Bansal
January 18th, 2020

## Human Activity Monitoring System

### Domain Background

With the advent and now ubiquitous presence of smart devices, such as cellular phones and fitness watches, around us an exciting new area of data mining research and data mining applications have been opened up. These devices have powerful sensors like GPS sensors, audio sensors, image sensors, direction and acceleration sensors like accelerometers and gyrometer, which help in recording highly accurate and real time. The field of research for this project is Wireless Sensor Data Mining(WISDM).

Recording and then analysing the data received from these sensors while performing daily day to day activities like walking, running, eating etc can help in creating a daily activity profile of a person. There is a possibility of providing a highly personalized and customized experience of smart devices like phones and watches. Also it can be determined if the user is following a healthy exercise routine or not.

### Problem Statement

This is a multi-class classification and a supervised learning problem. The idea is to build a system (model) which takes inputs from sensors like accelerometers and gyrometer.present in watches and mobile phones. The goal is to create a system which has good accuracy in terms of predicting the user activity.

### Datasets and Inputs

The dataset has been taken from UCI machine learning repository website, WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set . It is free to download. The data is present both as raw data ( time series dataset) and also as transformed labelled dataset. I will be using the transformed labelled dataset(ARFF files) . There are in total **18** labelled classes(Activities).

**Input Data fields**

- Raw Data Fields

1. **subject-id:** value from 1600- 1650 that identifies one of the 51 test subjects
2. **activity-code**: character between 'A' and 'S' (no 'N') that identifies the activity. The mapping from code to activity is provided in the activity_key.txt file and in the dataset description document.
3. **timestamp:** Unix time (integer)
4. **x**: represents the sensor reading (accelerometer or gyroscope) for the x dimension
5. **y**: represents the sensor reading (accelerometer or gyroscope) for the y dimension
6. **z**: represents the sensor reading (accelerometer or gyroscope) for the z dimension

- Transformed Data Fields

1. **ACTIVITY**: specifies the activity performed using one of the activity codes from X{0-9}; Y{0-9}, Z{0-9}: These 30 features collectively show the distribution of values over the
2. **x, y, and z axes**. We call this a binned distribution. For each axis we determine the range of values in the 10s window (max – min value), divide this range into 10 equal-sized bins, and then record the fraction of values in each bin.
3. **{X,Y,Z}AVG**: Average sensor value over the window (per axis).
4. **{X,Y,Z}PEAK**: Time in milliseconds between the peaks in the wave associated with most activities. Heuristically determined (per axis).
5. **{X,Y,Z}ABSOLDEV**: Average absolute difference between each of the 200 readings and the mean of those values (per axis)
6. **{X,Y,Z}STANDDEV**: Standard deviation of the 200 values (per axis)
7. **{X,Y,Z}VAR**: Variance of the values (per axis)
8. **XMFCC{0-12}, YMFCC{0-12}, ZMFCC{0-12}:** MFCCs represent short-term power spectrum of a wave, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. There are 13 values per axis.
9. **{XY, XZ, YZ}COS**: The cosine distances between sensor values for pairs of axes (three pairs of axes).
10. **{XY, XZ, YZ}COR:** The correlation between sensor values for pairs of axes (three pairs of axes).
11. **RESULTANT**: Average resultant value, computed by squaring each matching x, y, and z value, summing them, taking the square root, and then averaging these values over the 200 readings.
12. **Class**: This is a very different feature than the rest. It simply is set to the subject-id.

Target Variable is **ACTIVITY,** whose value is again mapped to a particular activity.

**Solution Statement**

The solution will be predicting the correct activity, out of the 18 activities listed, in the test dataset. The aim would be to achieve high accuracy. There are 43 features initially which is fairly large for a classification problem. I will do feature selection to have a limited set of features to determine the activity label.

For training models I will use AWS sagemaker inbuilt algorithms like Linear Learner and also some custom algorithms like logistic regression, decision trees, nearest-neighbors, and SVM since this is a classification problem. Finally I will select the best model for this problem and fine tune parameters to get the best accuracy.

**Benchmark Model**

For this problem, the benchmark model will be Linear Learner model with predictor_type as 'multiclass_classifier'. I will try to beat its performance with other algorithms.

**Evaluation Metrics**

For each of these classifications model, prediction results are evaluated based on **Accuracy** and **F1 score.** The model with highest accuracy wins.

**Project Design**

The training files are split across folders like phone and watch and inside them gyrometer and accelerometer readings. The data needs to be read from all the folders and consolidated. Before even start training models, I will first take a glimpse of the data to see what the shape and is and how they are formatted. I may perform some graph visualization for better understanding of the data distribution

I will check for any missing data values. Since in this case there are too many features, feature selection is required. I have three methods for feature selection in mind: Univariate Selection, Feature Importance, Correlation Matrix with Heatmap.I plan to evaluate these methods.

I plan to use Amazon Sagemaker from model training and deployment. I would be using some inbuilt algorithms of Sagemaker and also some other custom algorithms. I plan to use sckit-learn library and compare 3-4 different models, cross validate them and choose the one with high accuracy.

Because this is a multi-class classification problem, a few approaches in my head would be KNN, Decision trees, SVM, etc.

I expect to spend 60% of the time on data cleaning and feature selection and 40% of the time on training models and tweaking parameters. The final accuracy will be calculated against the test data sample.

**Reference**


- https://archive.ics.uci.edu/ml/index.php
- http://www.cis.fordham.edu/wisdm/includes/files/sensorKDD-2010.pdf
- https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e
- Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living. IEEE Access, 7:133190-133202, Sept. 2019.