BDPP Project Report: Human Activity Recognition using Smartphones Dataset

Lingyun Cheng

May 22, 2020

1 Introduction

Human Activity Recognition (HAR) is a new and appealing research field as the demand for understanding human activities have grown in the health-care domain. HAR aims to identify the actions carried out by a person given a set of observations of him/herself and the surrounding environment. These actions can be recognized by exploiting the information retrieved from various sources such as environmental or body-worn sensors. The modern smartphone comes equipped with a variety of sensors, from motion detectors to optical calibrators. The data collected by these sensors is valuable for better aligning the applications on the phone with the user's lifestyle. In this project, I used data collected from motion sensors in smartphones to build a model that identifies the type of human activities[3]. The end goal is to create a model that can classify the activities accurately without sacrificing the limited computational resources available on a single phone.

2 Data Collection and Preparation

2.1 Data Source

The dataset is from UCI Machine Learning repository and built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The system uses an embedded accelerometer and gyroscope to collect time-series signals, from which 3-axial linear acceleration and 3-axial angular velocity are captured. The sensor signals were preprocessed to generate features in both the time and frequency domain[4]. Table 1 shows the details of the signals.

Name	Time	Frequency
Body Acc	1	1
Gravity Acc	1	0
Body Acc Jerk	1	1
Body Angular Speed	1	1
Body Angular Acc	1	0
Body Acc Magnitude	1	1
Gravity Acc Mag	1	0
Body Acc Jerk Mag	1	1
Body Angular Speed Mag	1	1
Body Angular Acc Mag	1	1

Table 1: Time and frequency domain signals obtained from the smartphone sensors.

read data in the cloud IBM Watson Studio The dataset available in the UCL machine learning repository is in .txt file, when I uploaded it and read it

in the IBM Watson Studio with pyspark setting, I can not load it in spark data frame, as shown below:

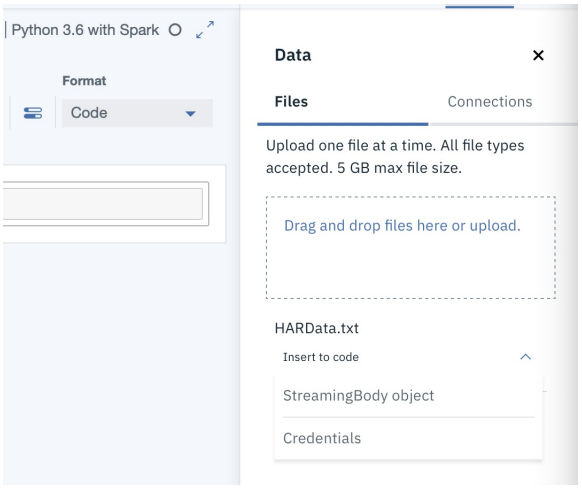


Figure 1: Problem of loading data in cloud

Then I convert it into .csv file so I can insert spark data frame.

2.2 Data Overview

The dataset consists of 10299 instances, each instance has 561 attributes, a id number of subject, and one label. I checked for missing values in all features but found none. Each feature has been standardized in the value range [-1,1].

2.3 Exploratory Analysis

visualize data in cloud I did not find the free visualization package available in the cloud with pyspark setting, so I convert the data into pandas frame and use matplotlib to do visualization analysis.

Target The last attribute 'Activity' is the target. There are six categories of activities: 'laying', 'standing', 'sitting', 'walk', 'walkup' and 'walkdown'. The distribution of activities is shown in Figure 1.

From Figure 2 we can see the relatively equal distribution of human motions captured in this experiment,which indicates there is no class imbalance. 'Laying' occurred at the highest frequency compared to other 5 activities.

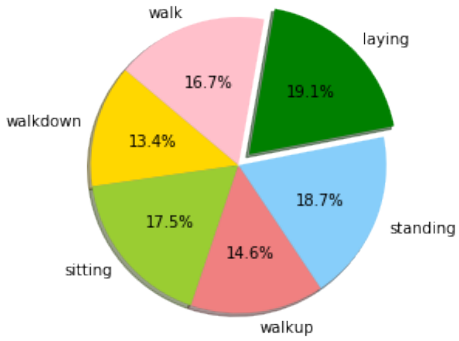


Figure 2: Distribution of Activities

Sensors Previous section introduced that the data is collected by sensors embedded in smartphones, I am curious about how many features are generated by specific sensors, so I plot the counts of features for different types of sensors.

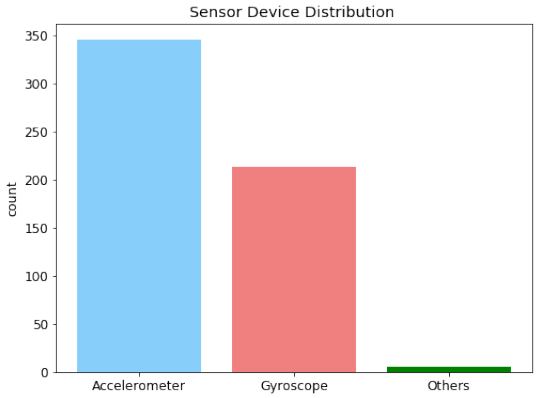


Figure 3: Distribution of Sensor Devices

It is obvious (Figure 3) that three types of devices are used for collecting signals and most features are generated using accelerometer and gyroscope.

Correaltion The dataset has 561 features and most features are generated from the signal sets shown in Table 1, so my hypotheses are most motions will be highly correlated.
As Figure 4 shown, most features are highly correlated with each other so dropping these highly correlated features is a good idea since the same information is stored in some other features with high correlations to a group of them.

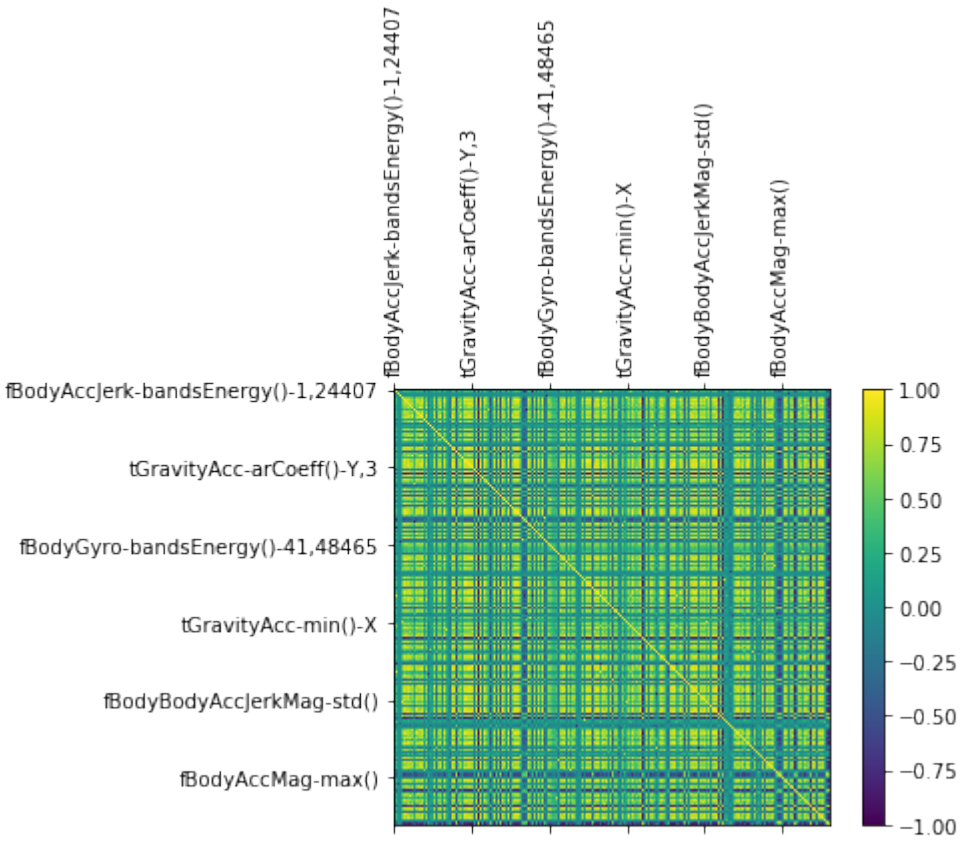
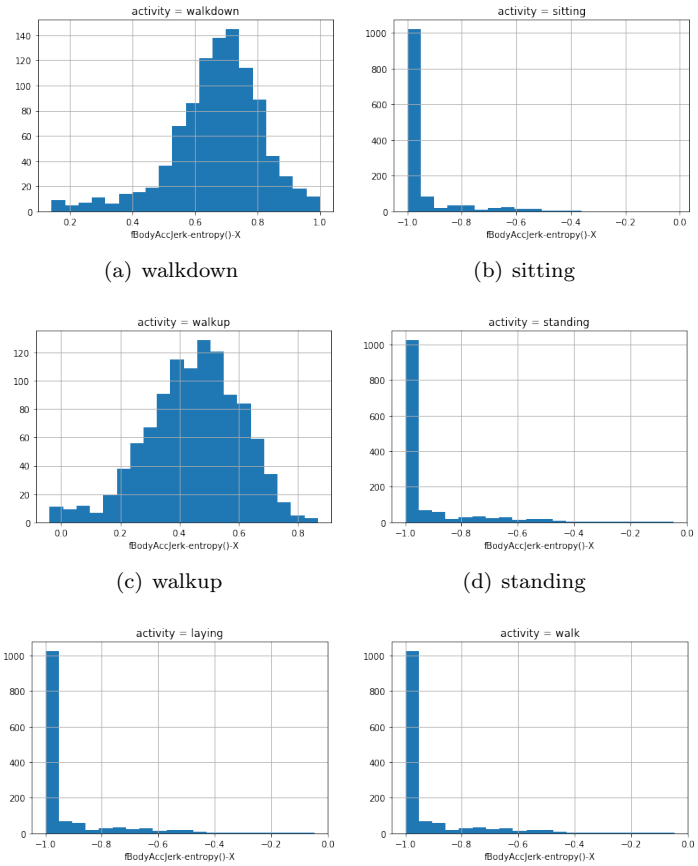


Figure 4: Correlation matrix of all features

Variance Given the large set of features of the dataset, the variables which have zero or low variance can be removed since it will eventually have a smaller impact on the classification model itself. I sorted the variances of all variables and found the feature 'fBodyAccJerk-entropy()-X' had the highest variance.

As seen from Figure 5, distribution of feature 'fBodyAccJerk-entropy()-X' changed for different classes of targets, which indicates this feature influence the classification result.



(a) walkdown (b) sitting

(c) walkup (d) standing

(e) laying (f) walk

Figure 5: Distribution of 'fBodyAccJerk-entropy()-X' across six activities

Cluster Since the target in this project has six categories, I want to check if the variation in the feature space can seperate the data into several groups. I did k-means clustering and determine the optimal number of clusters using elbow method, Figure 6 displayed the k-means score against number of clusters:

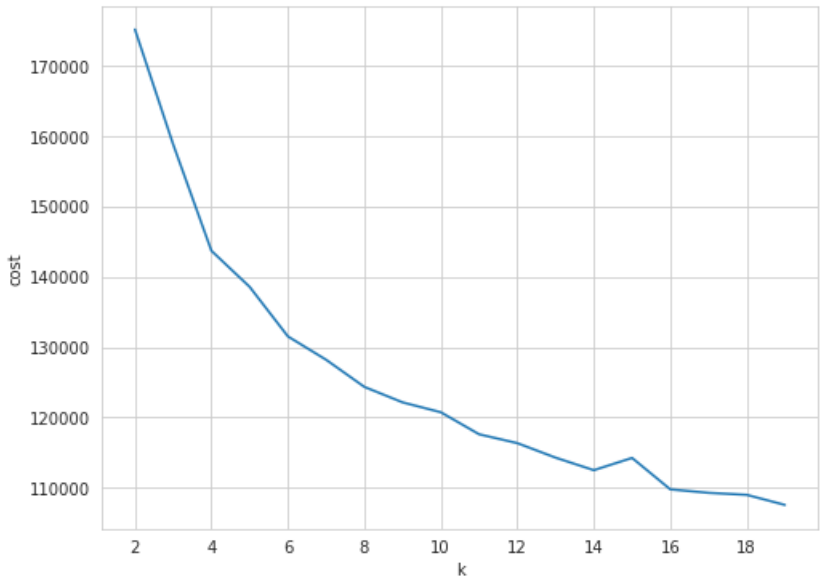


Figure 6: K-means score vs. number of clusters

From Figure 6, we can choose 4 or 6 as optimal number of clusters, so I suppose that the data can be separated into six classes but there might be two classes in them that are hard to classify.

3 Data Preprocessing

The attribute 'subject' represents the identifier of the action performers which is needless for the classification, so it is removed from the dataset. All features have been standardized so they can be used to train models directly. The target 'activity' is categorical attribute, so I converted it to numerical column as shown in Table 2.

activity	label
'laying'	0
'standing'	1
'sitting'	2
'walk'	3
'walkup'	4
'walkdown'	5

Table 2: Labels of activity

4 Dimension Reduction

Given that the dataset has a high dimension, I reduce the dimension to capture the dataset's structure and understand the distinctions between the labels better. I implemented two well-known algorithms: principal component analysis (PCA) and t-distributed stochastic neighbor embedding analysis (t-SNE)[5]. Figure 7 is the 2D plot of the dataset with PCA method and Figure 8 is the 2D plot of the dataset with t-SNE.

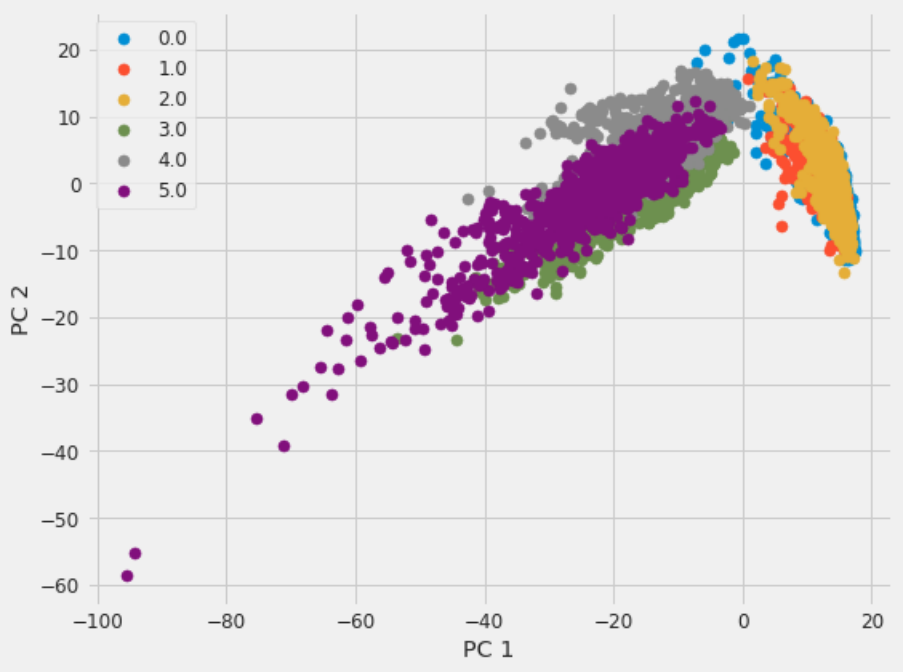


Figure 7: 2D projection of the data with PCA

Figure 7 shows the distribution of two principal components obtained by PCA, although these components have large variance compared to other varibales, PCA can only visualize the structure data through the linear subspace.

Figure 8 shows dataset after processed using the t-SNE algorithm[2], which can capture non-linear paths.

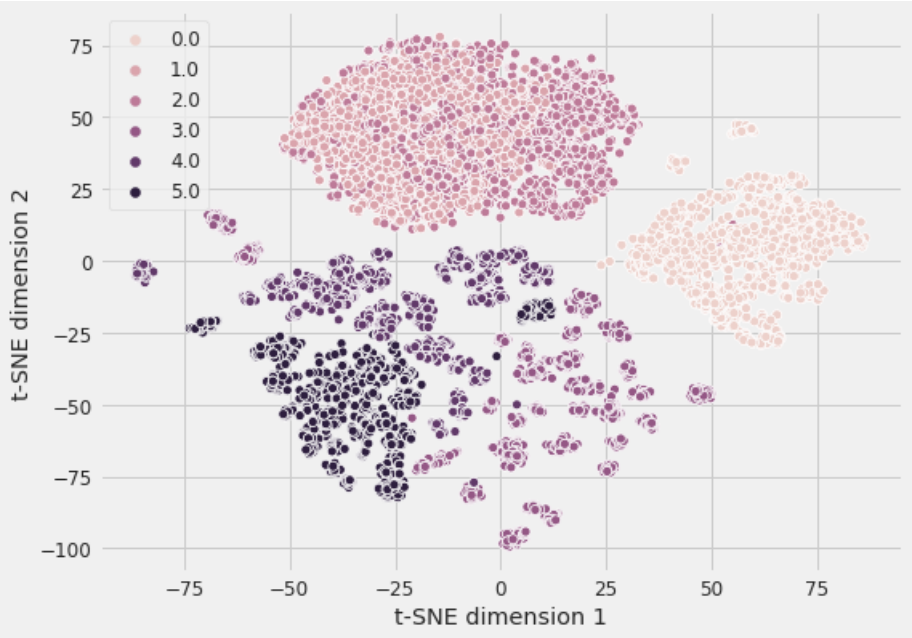


Figure 8: 2D projection of the data with t-SNE

Looking at Figure 7 and Figure 8, we can see both algorithms effectively distinguish between activities of motion (walk, walk up, walk down) and static activities (laying, standing, sitting), and each of the activities are well represented by a cluster. Within all activities, standing(label 1) and sitting(label 2)overlap most, which suggests that distinguishing between these two activities may pose a problem for classification models.

5 Models

Three models are used to predict the human activity: Logistic regression, Random forest, and Support Vector Classifier.

Since the target has six labels so I implemented multiclass classification through a One-Vs-All (OVA) approach. 'OneVsRest', also known as 'One-vs-All', can perform binary classification on each of the k categories. The prediction is done by evaluating each binary classifier, and the index of the most reliable classifier will be output as a label.

5.1 Logistic regression

Logistic regression is simple and runs quickly, so it serves as a baseline in this analysis. Logistic regression acted as a probabilistic discriminate classifier.

5.2 Random Forest

Random Forest use multiple decision trees which are built on separate sets of examples drawn from the dataset. In each tree, a subset of all the features can be used. By using more decision trees and averaging the result, the variance of the model can be greatly lowered. One typical tunable parameter is called 'numTrees', which represents the number of trees in the forest. I initially trained a random forest model with default setting and then tuned the parameter "numTrees", the performances are compared in the following section.

5.3 Support Vector Machines

A support vector machine constructs a hyperplane to separate each different pair of labels in a high-dimensional space, which can be used for classification, regression, or other tasks. LinearSVC in Spark ML supports binary classification with linear SVM. I extended a standard LinearSVC using 'OneVsRest' approach.

6 Evaluation and Results

To prepare for the model training, the dataset split into train set(70%) and test set(30%). The models are evaluated using two metrics: accuracy and confusion matrix[1]. The results are listed below.

6.1 Three models trained by all features

Three classifiers with default setting and trained by 561 features. Table 3 display the prediction accuracy of each models on test data:

Models	Test Accuracy
Logistic regression	90.67%
Random forest	94.02%
LinearSVC	95.79%

Table 3: Accuracy of models on test data

Each model achieves high accuracy(higher than 90%), but logistic regression performed the worst. It is reasonable because in most cases, the tree-based model performs better than linear classifiers.

Linear kernel SVM did perform slightly better than the random forest, one reason could be the parameter of the tree-based model has not been tunned, another reason might be the data is linear separately as shown in 2D projection (Figure 7 and Figure 8), so the SVC with the linear kernel can better separate the classes.

Next, to examine the models' accuracy in classifying each activity, I computed the confusion matrix, which presented the results of predicted labels and true labels.

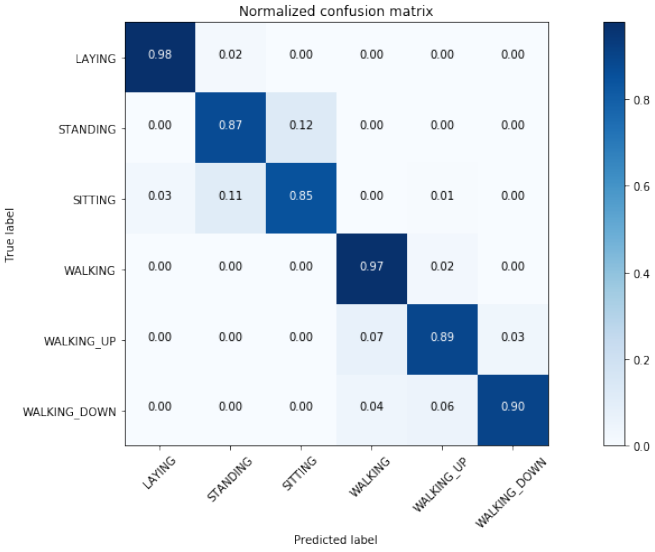


Figure 9: Confusion Matrix of Logistic regression

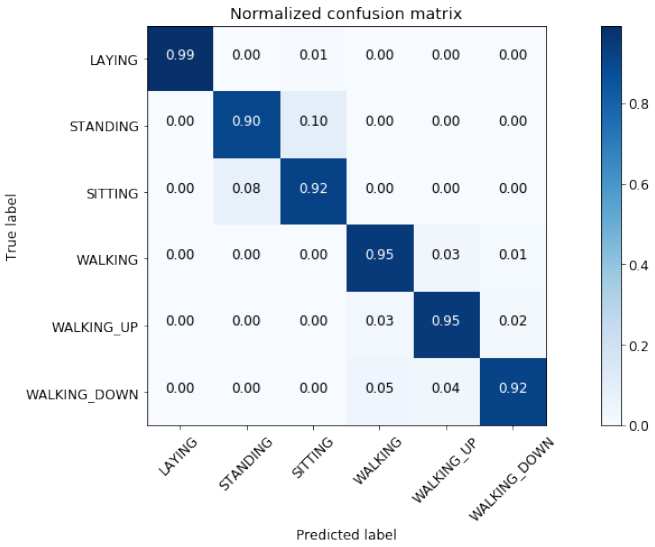


Figure 10: Confusion Matrix of Random Forest

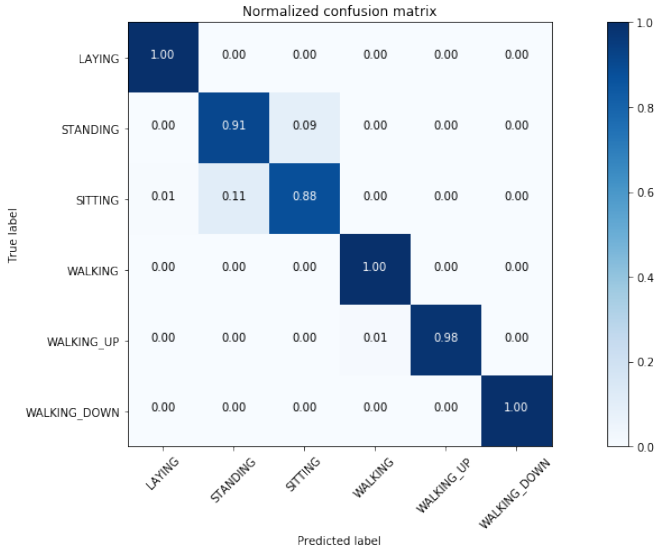


Figure 11: Confusion Matrix of linear SVC

From Figure 9, Figure 10 and Figure 11, we can see the sitting posture is often misclassified, whose misclassification rate range from 0.08 to 0.15, and almost all errors are mistakenly identified as standing. The results confirmed my assumption based on 2d protection visualization that activities can be perfectly divided into two categories: dynamic activities and static activities. However, activities in the same group are more likely to be mistaken for each other. In addition, errors mainly occur during the transition from standing to sitting position.

6.2 Optimization of Random Forest

In previous part, I assume that the reason linearSVC is better than random forest is that the default parameters cannot make the random forest model achieve the best performance. So in this step, I aims to improve the random forest by selecting the best parameter values and reduce feature size in case over-fitting.

6.2.1 Feature selection

To reduce overfitting and improve the generalization of random forest model, I did feature selection to reduce the size of features. Only the features that are important for classification are selected to train a model. I calculated the importance scores of each feature for random forest model with 18 trees and Figure 12 displayed the top 20 features with high scores.

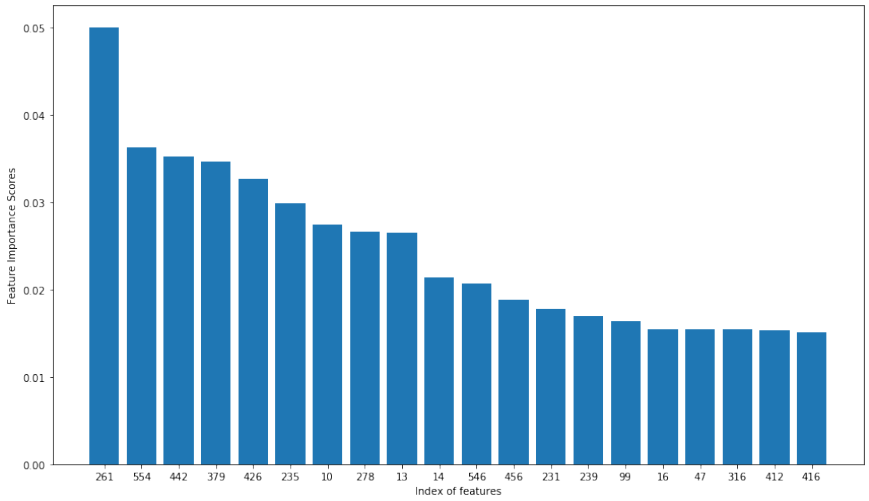


Figure 12: Top 20 important features for random forest model with 18 trees

Then I trained model on feature sets with different size to find optimal number of features that can improve the accuracy of the model. Figure 13 shows the performance of random forest model with various number of features.

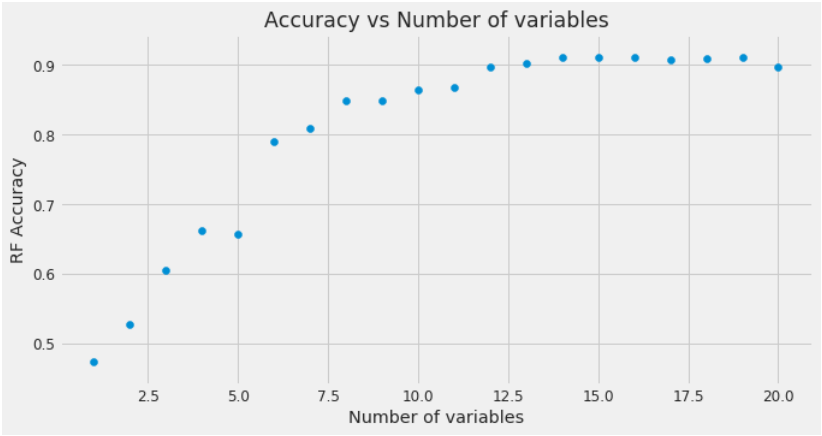


Figure 13: Number of Features vs. Accuracy for Random Forest

As shown in Figure 13, RF classifier achieved 80% accuracy with 7 most important variables and keep accuracy higher than 90% with 12 or more important features.

Next, I tuned the parameter 'number of trees' of the RF model. Figure 14 shows the accuracy of RF model against different number of trees.

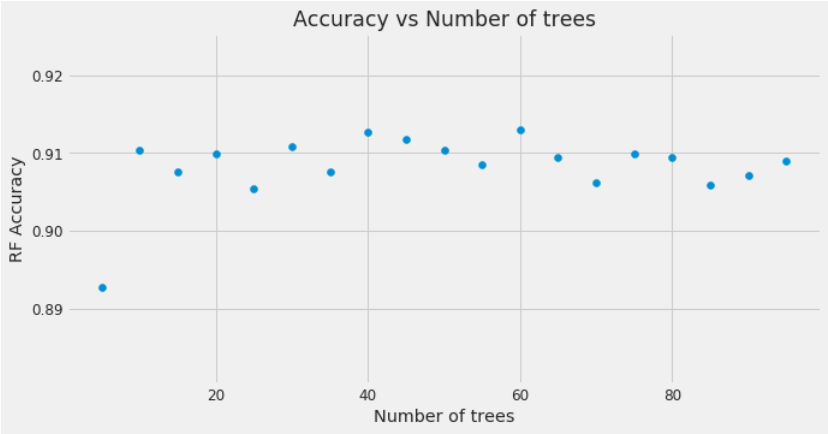


Figure 14: Number of Trees vs. Accuracy for Random Forest

From Figure 14, we can see the model perform relatively stable when the number of trees beyond 10.

Finally, the optimal number of trees and feature size I choose are 60 and 15 respectively, the performance of optimal RF model is shown as below:

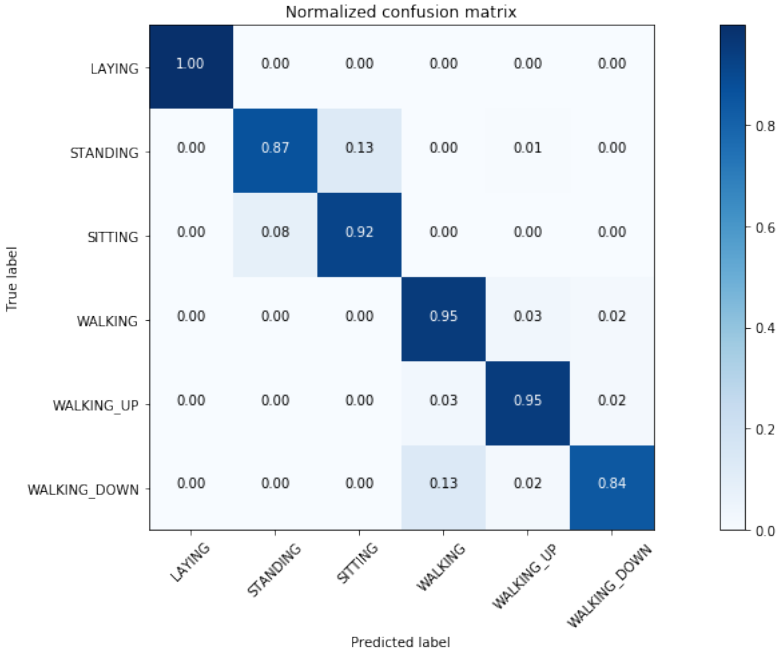


Figure 15: Confusion Matrix of Optimal Random Forest

The optimal RF model achieved 92.34% accuracy and the misclassification rate of 'laying' decreased to zero. However the ability of classifying 'standing' and 'sitting' did not improve.

7 Discussion

Overall, three classifiers (Logistic regression, Random forest, and Support Vector Classifier) achieved relatively high performance (all above 90% accuracy). The best classification accuracy in this experiment was 95.79%, which is achieved by SVM with a linear kernel. All the models performed not effectively when distinguish 'sitting' and 'standing'. In future work, perhaps having additional features that can describe the difference between these two postures could help to improve the performance of classifiers.

References

[1] https://medium.com/usf-msds/choosing-the-right-metric-for-evaluating-machine-learning-mo

[2] t-sne python example. https://towardsdatascience.com/t-sne-python-example-1ded9953f26.

[3] Ucl machine learning repository. https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones. Accessed: 2012-12-10.

[4] Luca Oneto Xavier Parra Davide Anguita, Alessandro Ghio and Jorge L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones., April 2013.

[5] Fjoralba Shemaj. Nicholas Canova. Human activity recognition using smartphone sensor data., December 2016.