# Decent Logger: An Activity Positioning Prediction System

CS 205: Health Analytics

Project Report


| | |
|---|---|
| Anoosha Sagar | 605028604 |
| Devanshi Tahilramaney | 104946297 |
| Pujita Anand Tibrewala | 305029365 |
| Saloni Goel | 604944130 |

**Introduction** **3**

**Literature Review** **4**

**Methodology Followed** **6**

- Dataset Collection 6
- Preprocessing & Feature Extraction 6
  - Data Trimming 7
  - Overlapping Windows & Non-Overlapping Windows 7
  - Feature Selection & Extraction 7
    - Time Domain Features 7
    - Frequency Domain Features 9
    - Magnitude based features 9
- Classification Models 10
- Evaluation Metrics 11

**Implementation Details** **11**

**Results** **12**

- Results for Combined Accelerometer & Gyroscope Sensors 12
- Results Using Only Accelerometer Sensor 17
- Results Using Only Magnitude Based Features in the Accelerometer Sensor 21

**Observations & Key Takeaways** **25**

**Challenges** **25**

**Conclusion** **26**

**References** **26**

## Introduction

In this project, we implemented an activity monitoring system for the classification of user activities such as:

- Walking
- Sitting
- Standing
- Laying down

Which were captured using a smartwatch provided to us in the course. Various phases were involved in the fulfilment of this project ranging from data collection, data preprocessing, feature extraction, and model selection and testing.

Different classification models such as Logistic Regression, Support Vector Machine Classifier, Random Forest Classifier, Decision Tree Classifier, etc. performing hyperparameter tuning which resulted in good results which showcased the strength of our classification models.

## Literature Review

Previous studies involving automatic activity recognition for data collected from smart watch/wrist area have been surveyed.

Many past works have demonstrated 85% to 95% recognition rates for ambulation, posture, and other activities using only the acceleration data. Activity recognition has also been performed on acceleration data collected from the hip (e.g. [11, 12]) and from multiple locations on the body (e.g. [13, 14]). The energy of a subject's acceleration can discriminate sedentary activities such as sitting or laying down from moderate intensity activities such as walking or standing and vigorous activities such as running [15]. Recent work with 30 wired accelerometers spread across the body suggests that the addition of sensors will generally improve recognition performance [16].

Although the literature supports the use of acceleration for physical activity recognition, little work has been done to validate the idea under real-world circumstances. Most prior work on activity recognition using acceleration relies on data collected in controlled laboratory settings. Typically, the researcher collected data from a very small number of subjects, and often the subjects have included the researchers themselves. The researchers then hand-annotated the collected data. But, ideally, data would be collected in less controlled settings without any researcher supervision. Further, to increase the volume of data collected, subjects would be capable of annotating their own data sets. Algorithms that could be trained using only user-labeled data might dramatically increase the amount of training data that can be collected and permit users to train algorithms to recognise their own individual behaviours.

However, one recent study has shown that unsupervised learning can be used to cluster accelerometer data into categories that, in some instances, map onto meaningful labels [17]. The vast majority of prior work focuses on recognising a special subset of physical activities such as ambulation, with the exception of [18] which examines nine everyday activities. Interestingly, [18] demonstrated 95.8% recognition rates for data collected in the laboratory but recognition rates dropped to 66.7% for data collected outside the laboratory in naturalistic settings. These results demonstrate that the performance of algorithms tested only on laboratory data or data acquired from the experimenters themselves may suffer when tested on data collected under less-controlled (i.e. naturalistic) circumstances.

Prior literature demonstrates that forms of locomotion such as walking, running, and climbing stairs and postures such as sitting, standing, and lying down can be recognised at 83% to 95% accuracy rates using hip, thigh, and ankle acceleration. All past works with multiple accelerometers have used accelerometers connected with wires, which may restrict subject movement. Based on these results, this work uses data collected from five wire-free biaxial accelerometers placed on each subject's right hip, dominant wrist, non-dominant upper arm, dominant ankle, and non-dominant thigh to recognize ambulation, posture, and other everyday activities. Although each of the above five locations have been used for sensor placement in past work, no work addresses which of the accelerometer locations provide the best data for recognizing activities even though it has been suggested that for some activities that more sensors improve recognition [16]. Prior work has typically been conducted with only 1-2 accelerometers worn at different locations on the body, with only a few using more than 5 (e.g. [12,14,16]).

State-of-the-art methods for recognizing human activity using raw data from body worn accelerometers have primarily been validated with data collected. In this paper [8], they have taken up an existing classification system and added few additional features collected from both wrist and ankle worn devices. These new features increased the performance by 2.3% resulting the overall 88.5%.

In this paper[19], they presented a comparison of 14 methods to extract classification features from accelerometer signals. These were based on the wavelet transform and other well-known time- and frequency-domain signal characteristics. To allow an objective comparison between the different features, they used two datasets of activities collected from 20 subjects. The first set comprised three commonly used activities, namely, level walking, stair ascent, and stair descent, and the second a total of eight activities. Additionally, they compared the classification accuracy for each feature set across different combinations of three different accelerometer placements. The classification analysis has been performed with robust subject-based cross-validation methods using a nearest-neighbor classifier. The findings show that, although the wavelet transform approach can be used to characterize non-stationary signals, it does not perform as accurately as frequency-based features when classifying dynamic activities performed by healthy subjects. Overall, the best feature sets achieved over 95% intersubject classification accuracy.

A key component of activity detection using accelerometer and other sensor data is the selection of relevant features which make the task of classification more efficient and accurate. Raw sensory data is not directly fed into classification algorithms, instead preprocessing is performed where the data is trimmed from both ends to account for the transition from activation of a label to going to the natural state. Additionally, noise can be filtered out from the signal using a low pass filter [7], or a butterworth filter where the bandwidth of the signal is clipped and high frequency noise is removed [5]. After preprocessing, features are extracted at certain intervals which helps create windows of the data. These windows can be of varying sizes ranging from 4s, 8s to 10s and can either be overlapping or non overlapping [5, 4, 3]. Various insights about the shape, size, and distribution of the signal can be obtained by calculating various statistical features such as mean, median, standard deviation, interquartile range, zero crossings, kurtosis, skewness, and cross correlation [3]. Additional features can also be extracted from the frequency and time-frequency representation of the signal.

# Methodology Followed

## Dataset Collection

- The Training data was collected from 11 people. We collected about 90 minutes of data for a every type of label.
- The Testing data was collected from 2 people. We collected about 20 minutes of data for each label.
- The data collected was a mix of clean and freestyle data, i.e., the data for various labels was collected considering a laboratory setting where the activities are more distinct and slight intermingling of activities was not permitted and additionally freestyle data where activities of the users in a more natural setting was also captured.

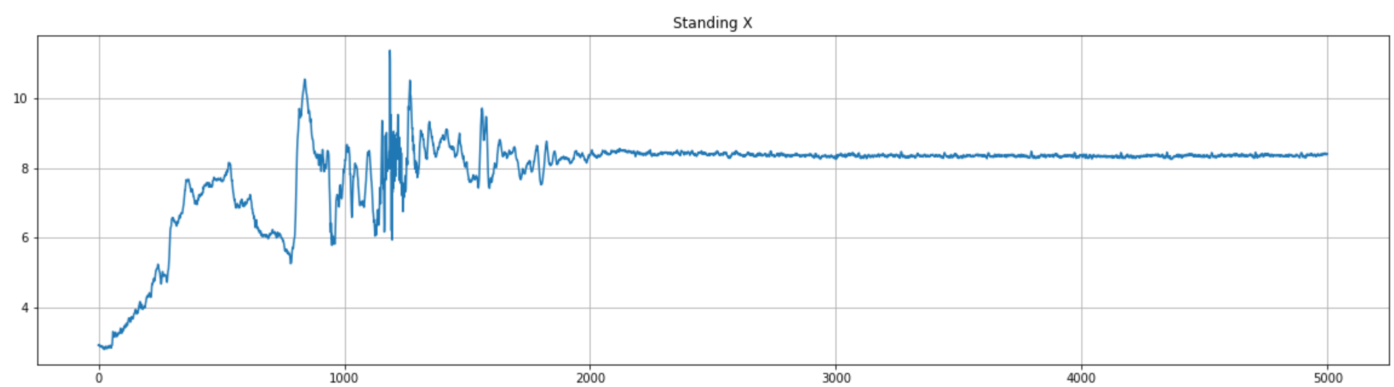
- The data captured was balanced, i.e., data for all the labels was collected for equal amount of time which would help our classifiers to learn better.


## Preprocessing & Feature Extraction

Raw data is not directly used for training the model. The following actions are needed to be performed before we start the classification task:

### Data Trimming

We need to trim the data in order to account for the time taken to activate and label and go to natural state. Hence the data is trimmed at the beginning of the data recording and at the end of the recording for a length of 10s.




### Overlapping Windows & Non-Overlapping Windows

The raw sensory data is combined into windows of different time durations. Each window contains information which describes the signal such as statistical features and frequency domain features.

The windows can be either overlapping or jumping/ non-overlapping. Many studies on activity prediction systems make use of either of them and achieve good results.

## Feature Selection & Extraction

Features are extracted from the sensor data which help us describe the shape, distribution, and nature of the signal which help make the classification process more efficient.

Features are extracted from the time domain and frequency domain of the signal.

**Time Domain Features**

| **Feature** | **Description** |
|---|---|
| *Mean* | Mean value of the array |
| *Variance* | Variance of the array |
| *Standard Deviation* | Standard Deviation of the array |
| *Median* | Median of the array |
| *Min* | Smallest value in the array |
| *Min Index* | Index of the component with the smallest magnitude |
| *Max* | Largest value in the array |
| *Max Index* | Index of the component with the largest magnitude |
| *Skew* | Measure of symmetry, or more precisely, the lack of symmetry |
| *Kurtosis* | Measure of the sharpness of the peaks |
| *Mean Absolute Deviation* | The average distance between each data value and the mean |
| *Autocorrelation* | The similarity between observations as a function of the time lag between them |
| *Interquartile Range* | Measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles |
| *Root Mean Square* | The square root of the arithmetic mean of the squares of the |

| | |
|---|---|
| | values |
| *Number of Zero Crossings* | Number of times the values change their sign |
| *Entropy* | Measure of disorder, using probabilistic parameters |
| *Cross Correlation* | Measure of similarity of two series as a function of the displacement between the two |

Mean, variance, standard deviation, median, mean absolute deviation, and interquartile range help describe the distribution of a signal. While mean and median both help determine the center of a distribution, both are used instead together because median is not sensitive to outliers where on the other hand mean is affected by outliers. Variance and standard deviation are both measures of spread of the distribution about the mean. Standard deviation is also influenced by outliers and thus is a good indicator for the presence of outliers. similarly , quartiles are also less affected by outliers.

Skewness and kurtosis are measures which help us determine the shape of the distribution. Skewness measures the relative size of the two tails and kurtosis is the measure of the combined sizes of the two tails. In general, data sets with high kurtosis tend to have heavy tails, or outliers whereas datasets with low kurtosis tend to have light tails or a lack of outliers.

Autocorrelation can help us find repeating patterns in a dataset and cross correlation is commonly used for searching a long signal for a shorter, known feature.

### Frequency Domain Features

Fast Fourier Transform (FFT) was applied to the signal to obtain the frequency domain representation of the signal. The following features were extracted from the FFT transform:

| **Feature** | **Description** |
|---|---|
| *Norm* | Magnitude of the FFT signal |
| *Energy* | The squared sum of spectral coefficients of the FFT representation |

**Magnitude based features**

Magnitude of the signal calculated as follows:

$$magnitude = \sqrt{x^2 + y^2 + z^2}$$

After calculating the magnitude of the signal, the following features were calculated on it:

- Mean
- Standard Deviation
- Min
- Max
- Skew
- Mean Absolute Deviation
- Entropy
- Variance
- Median
- Min Index
- Max Index
- Kurtosis
- Interquartile Range
- RMS

**Gyroscope Features**

The following features were calculated on the magnitude of the signal and for each of the axes respectively:

- Mean
- Variance
- Standard Deviation
- Median
- Min
- Max

## Classification Models

The classifiers used for prediction of activity are:

- **Random Forest Classifier** - A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).
- **Decision Tree Classifier** - Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model

that predicts the value of a target variable by learning simple decision rules inferred from the data features.

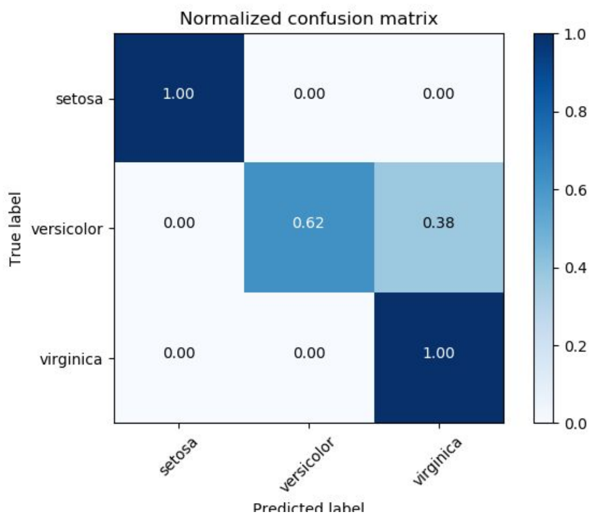
- **Multi Layer Perceptron Classifier (with 2 hidden layers)** - Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset, where $m$ is the number of dimensions for input and $o$ is the number of dimensions for output. Given a set of features $X = x_1, x_2, ..., x_m$ and a target $y$, it can learn a non-linear function approximator for either classification or regression.
- **K Neighbors Classifier** - Neighbors-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point. KNeighborsClassifier implements learning based on the $k$ nearest neighbors of each query point, where $k$ is an integer value specified by the user.
- **Logistic Regression** - Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.
- **Support Vector Machine Classifier** - Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables.

## Evaluation Metrics

For comparing the performance of our classifiers, the following metrics were chosen:

- **Accuracy**: Accuracy of a measurement system is the degree of closeness of measurements of a quantity to that quantity's true value. In this project, we have used accuracy to judge the performance of the various classifier models that have been used. Higher accuracy indicates that the model is performing better at classification.
- **Precision**: Precision refers to the closeness of two or more measurements to each other. This statistic was also used as an evaluation metric for our classifiers.
- **Confusion Matrix**: A confusion matrix is a specific table that allows visualization of the performance of an algorithm, typically a supervised learning one. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). Confusion matrices are

created in this project to depict how the classifier functions.




## Implementation Details

We developed our activity prediction system iteratively and implemented some enhancements at every iteration

- Iteration 1: Clean Data + Cross Validation
    - Initially only made use of accelerometer sensor
    - Initial features restricted to mean, variance, standard deviation, and median
    - 10 fold cross validation performed consistently giving high accuracy
- Iteration 2: Freestyle Data + Separate Training and Testing
    - Data collected was more natural, e.g., slight shuffling of feet while standing, hand movement while sitting, etc
    - Training data consisted of data from 3 people and testing data from 1 person
    - Drastic decrease in accuracy observed
- Iteration 3: Addition of New Features
    - Additional features such as skew, kurtosis, signal magnitude area, etc. Also included frequency domain features
    - Created a model combining accelerometer and gyroscope sensor data
    - Significant improvement in accuracy not observed
- Iteration 4: Larger Training Set and Hyperparameter Tuning
    - Increased the training dataset to contain data of 11 people and testing dataset to data of 2 people
    - Increased window size from 4 sec to 8 sec
    - Created a separate model containing only magnitude related features and added magnitude features to the existing accelerometer+gyroscope features

We created models for

a. Combined accelerometer and gyroscope sensors data
b. Accelerometer sensor data
c. Accelerometer sensor data (magnitude features only)

The preprocessing parameters are as follows:

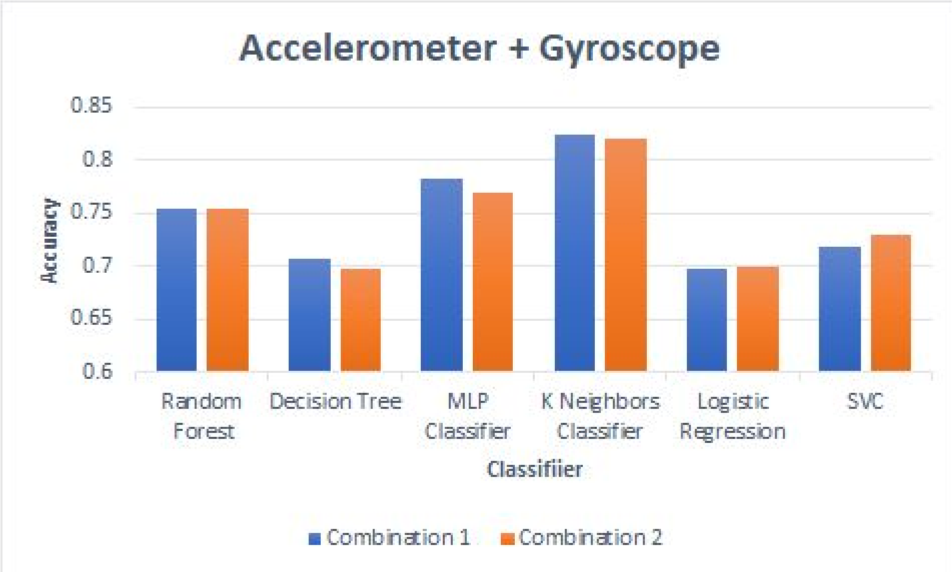
| | Combination 1 | Combination 2 |
|---|---|---|
| Trimming Cutoff | 10 sec | 10 sec |
| Window Size | 8 sec | 8 sec |
| Overlapping | True | False |
| Overlapping Factor | 0.5 | - |

# Results

## Results for Combined Accelerometer & Gyroscope Sensors

### Accuracy:

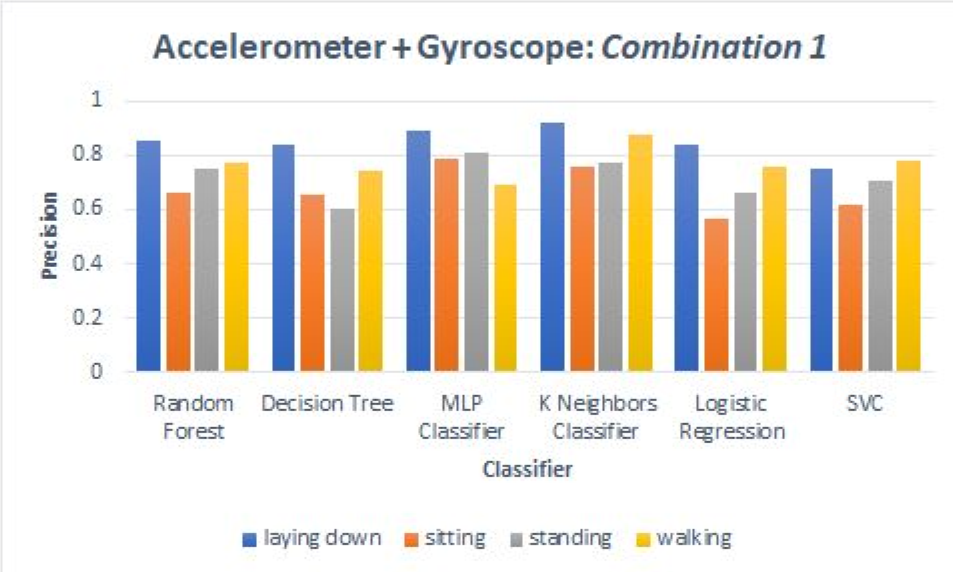
| | Combination 1 | Combination 2 |
|---|---|---|
| Random Forest | 0.753196931 | 0.753180662 |
| Decision Tree | 0.707161125 | 0.697201018 |
| MLP Classifier | 0.782608696 | 0.768447837 |
| K Neighbors Classifier | 0.823529412 | 0.819338422 |
| Logistic Regression | 0.696930946 | 0.699745547 |
| SVC | 0.718670077 | 0.730279898 |




## Precision:

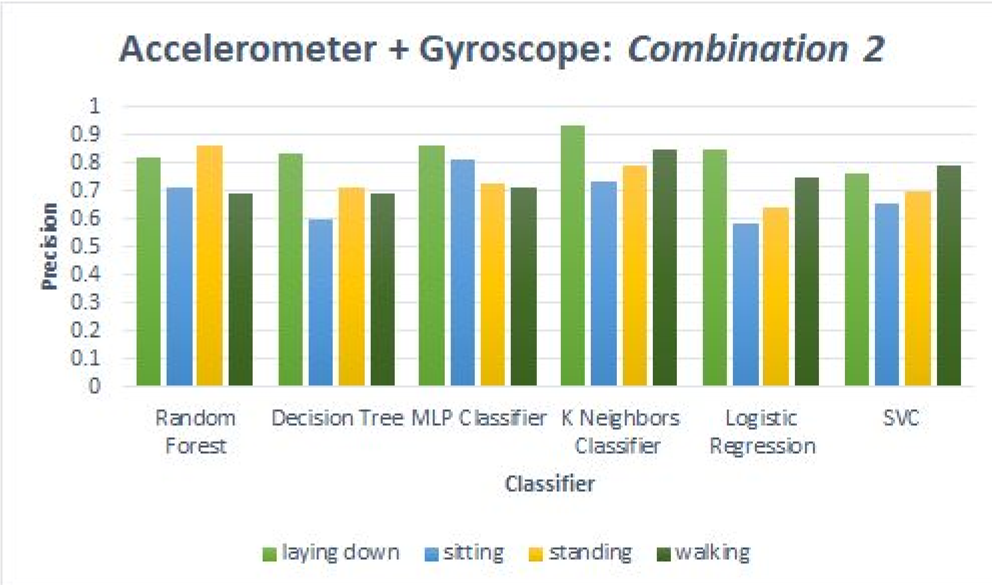
Combination 1

| | laying down | sitting | standing | walking |
|---|---|---|---|---|
| Random Forest | 0.85 | 0.66203704 | 0.75135135 | 0.77375566 |
| Decision Tree | 0.84065934 | 0.65789474 | 0.59913793 | 0.74537037 |
| MLP Classifier | 0.89240506 | 0.78421053 | 0.80978261 | 0.692 |
| K Neighbors Classifier | 0.92207792 | 0.76100629 | 0.77007299 | 0.87179487 |
| Logistic Regression | 0.83941606 | 0.5625 | 0.66222222 | 0.75877193 |
| SVC | 0.75294118 | 0.61437908 | 0.70464135 | 0.77927928 |

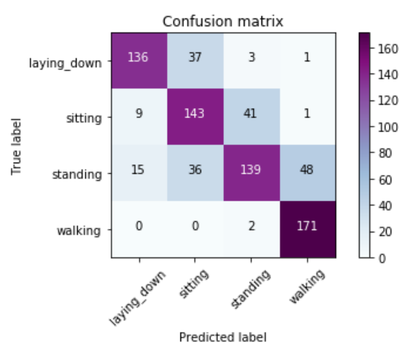
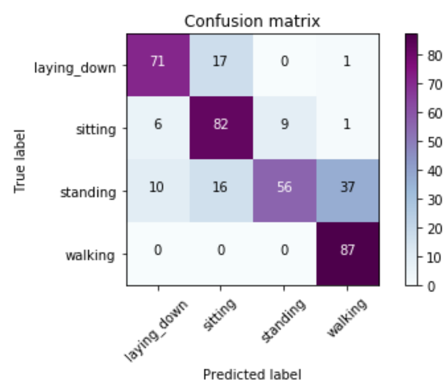
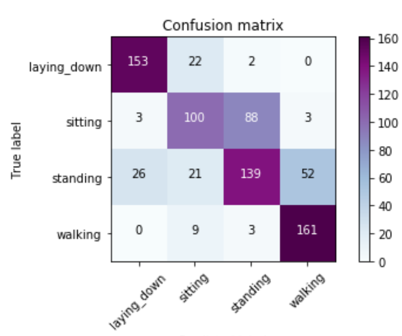
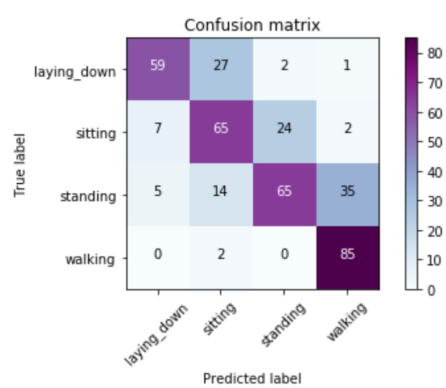



Combination 2

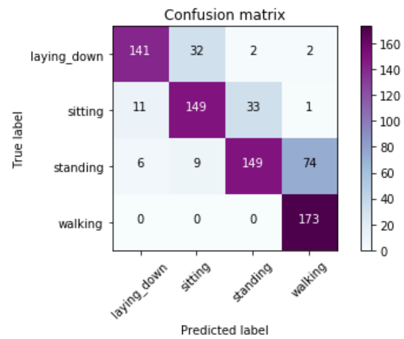
| | laying down | sitting | standing | walking |
|---|---|---|---|---|
| Random Forest | 0.81609195 | 0.71304348 | 0.86153846 | 0.69047619 |
| Decision Tree | 0.83098592 | 0.60185185 | 0.71428571 | 0.69105691 |
| MLP Classifier | 0.86206897 | 0.81012658 | 0.72380952 | 0.71311475 |
| K Neighbors Classifier | 0.93150685 | 0.73333333 | 0.79069767 | 0.85148515 |
| Logistic Regression | 0.84931507 | 0.58426966 | 0.64347826 | 0.75 |
| SVC | 0.76190476 | 0.65333333 | 0.7016129 | 0.79090909 |

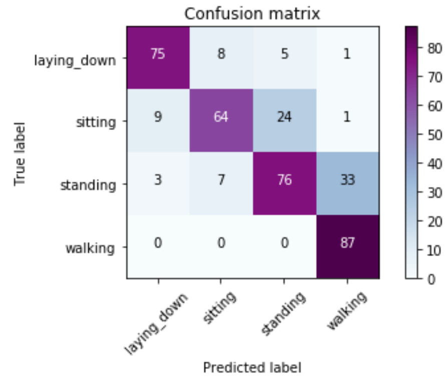



## Confusion Matrix:

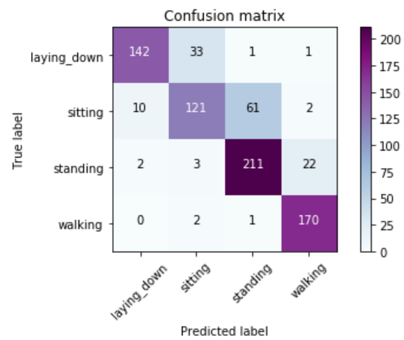
| | Combination 1 | Combination 2 |
|---|---|---|
| Random Forest |   |   |
| Decision Tree |   |   |

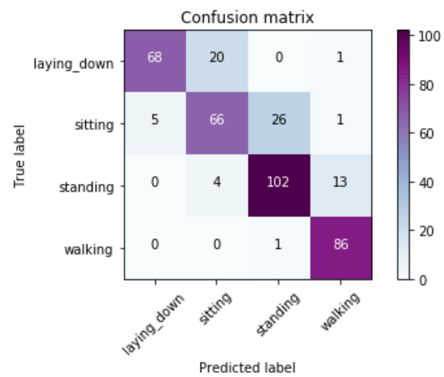
## MLP Classifier



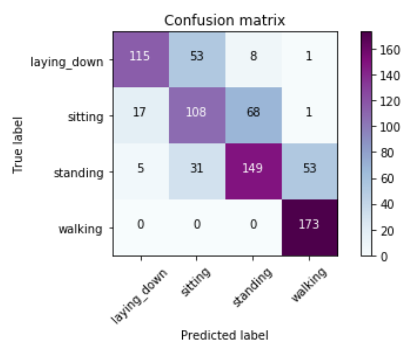



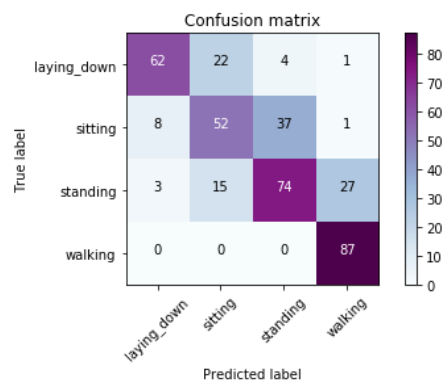

## K Neighbors Classifier



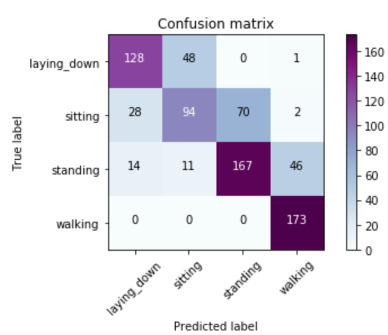



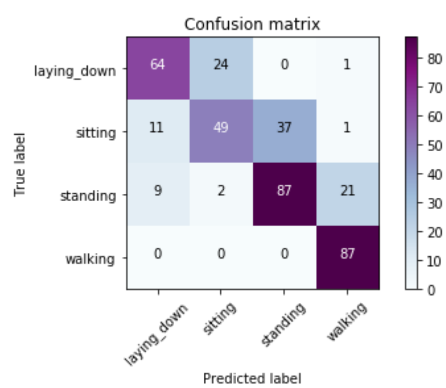

## Logistic Regression




## SVC

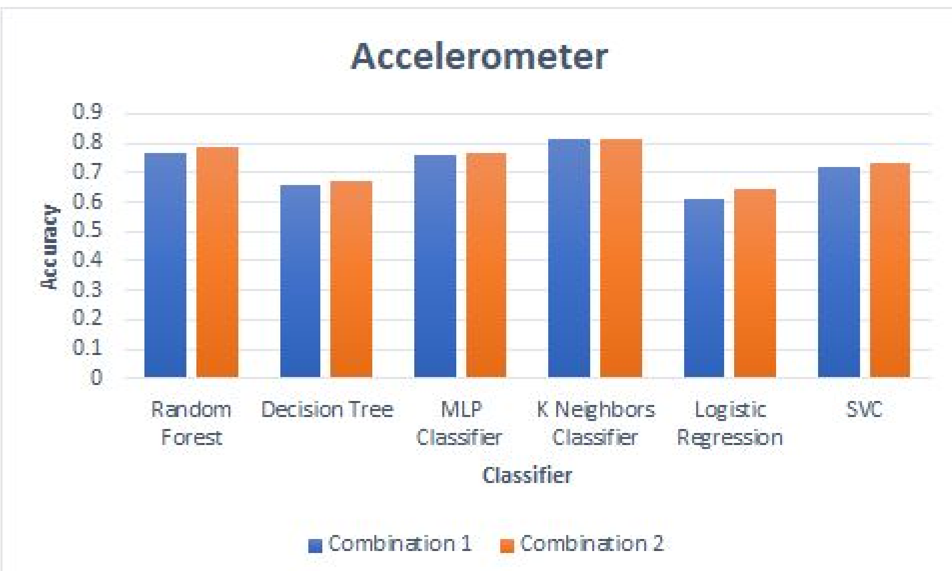


## Results Using Only Accelerometer Sensor

### Accuracy:

| | Combination 1 | Combination 2 |
|---|---|---|
| Random Forest | 0.767263427 | 0.783715013 |
| Decision Tree | 0.654731458 | 0.669211196 |
| MLP Classifier | 0.757033248 | 0.765903308 |
| K Neighbors Classifier | 0.814578005 | 0.814249364 |
| Logistic Regression | 0.609974425 | 0.646310433 |
| SVC | 0.719948849 | 0.730279898 |

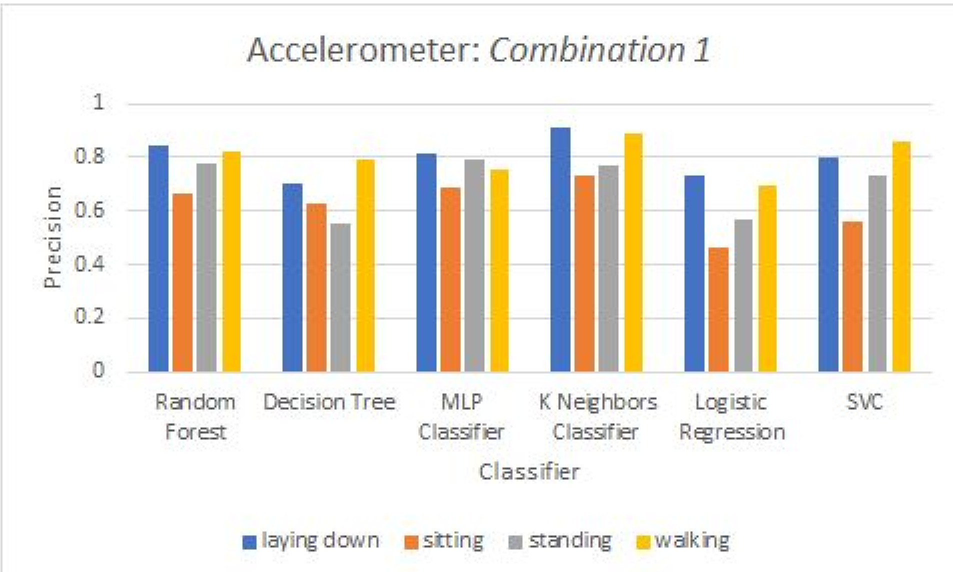



### Precision:

Combination 1

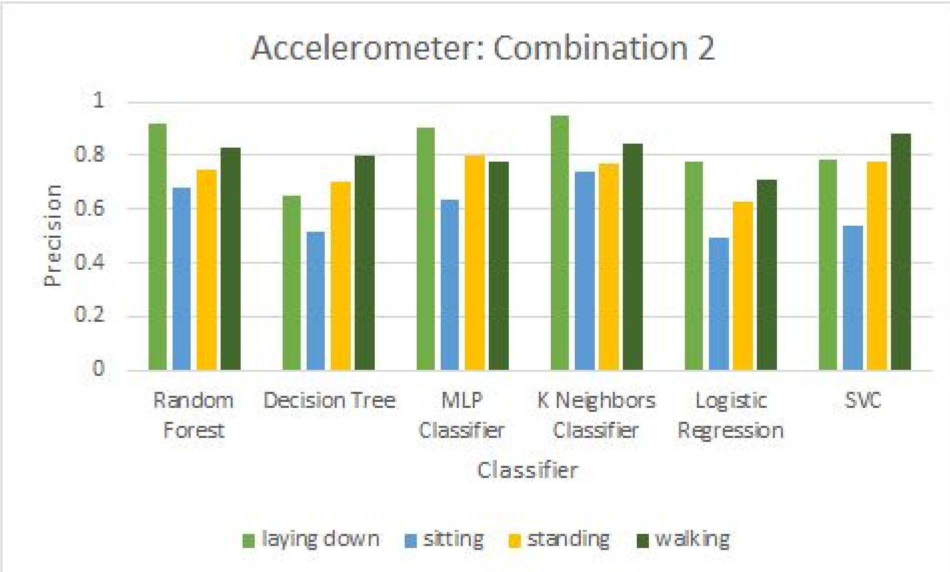
| | laying down | sitting | standing | walking |
|---|---|---|---|---|
| Random Forest | 0.8452381 | 0.66666667 | 0.77604167 | 0.82122905 |
| Decision Tree | 0.70351759 | 0.62857143 | 0.55021834 | 0.79166667 |

| | | | | |
|---|---|---|---|---|
| MLP Classifier | 0.81333333 | 0.68627451 | 0.78846154 | 0.75454545 |
| K Neighbors Classifier | 0.9109589 | 0.73099415 | 0.76785714 | 0.88648649 |
| Logistic Regression | 0.73529412 | 0.46534653 | 0.56521739 | 0.69458128 |
| SVC | 0.7983871 | 0.55744681 | 0.73221757 | 0.85869565 |

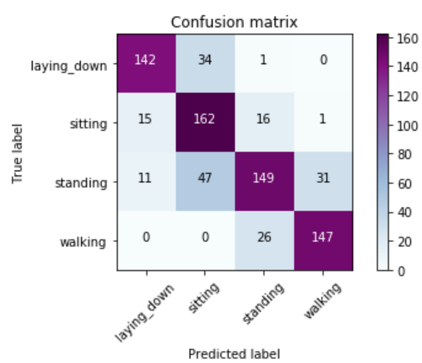
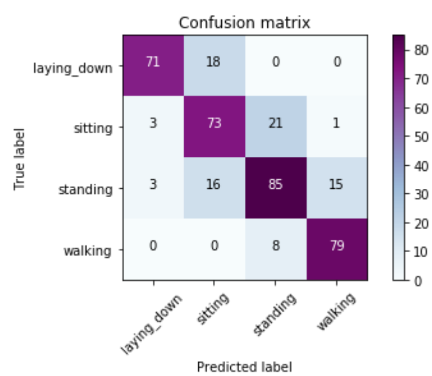
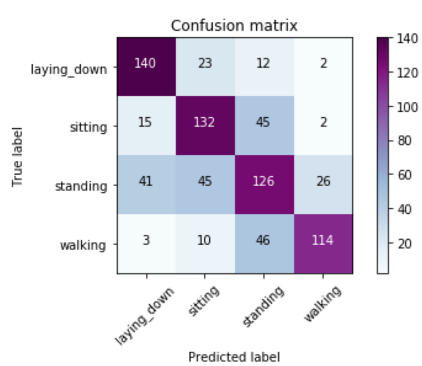
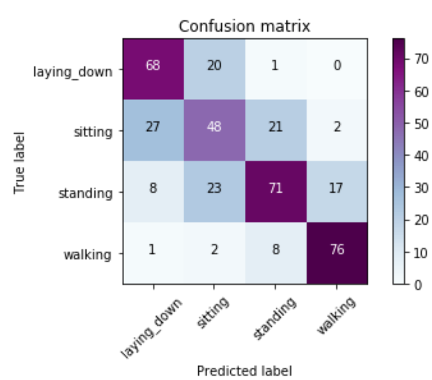



Combination 2

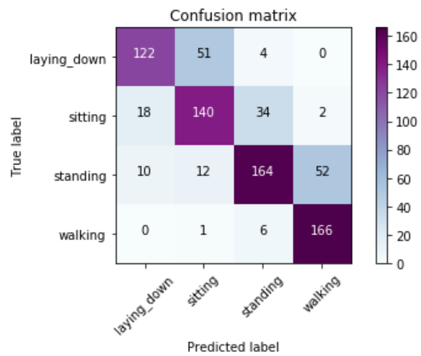
| | laying down | sitting | standing | walking |
|---|---|---|---|---|
| Random Forest | 0.92207792 | 0.68224299 | 0.74561404 | 0.83157895 |
| Decision Tree | 0.65384615 | 0.51612903 | 0.7029703 | 0.8 |
| MLP Classifier | 0.9 | 0.63636364 | 0.79807692 | 0.77981651 |
| K Neighbors Classifier | 0.94520548 | 0.73863636 | 0.77205882 | 0.84375 |
| Logistic Regression | 0.77777778 | 0.4952381 | 0.62637363 | 0.70689655 |
| SVC | 0.78333333 | 0.54098361 | 0.77586207 | 0.88421053 |

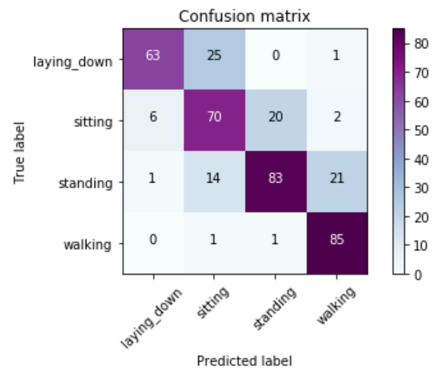



## Confusion Matrix:

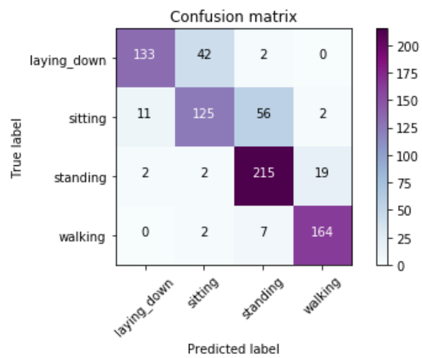
| | Combination 1 | Combination 2 |
|---|---|---|
| Random Forest |   |   |
| Decision Tree |   |   |

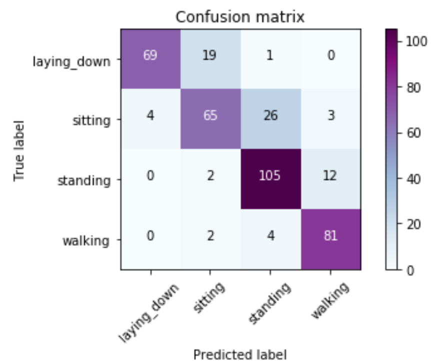
MLP Classifier



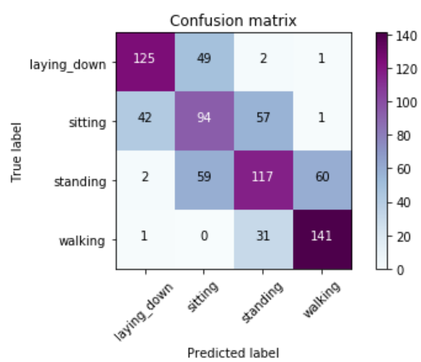



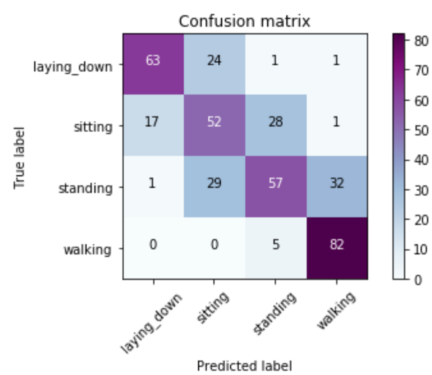

K Neighbors Classifier



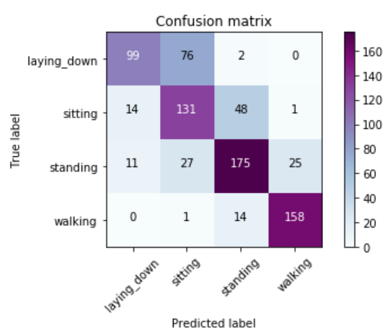



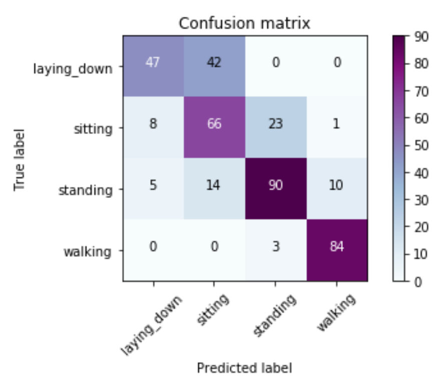

Logistic Regression




SVC

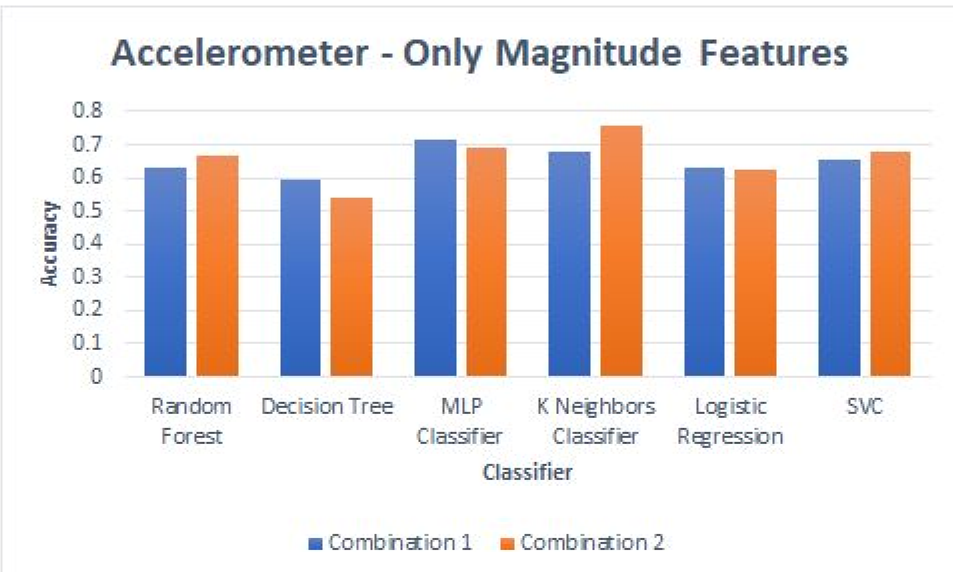


## Results Using Only Magnitude Based Features in the Accelerometer Sensor

### Accuracy:

| | Combination 1 | Combination 2 |
|---|---|---|
| Random Forest | 0.631713555 | 0.664122137 |
| Decision Tree | 0.592071611 | 0.541984733 |
| MLP Classifier | 0.71483376 | 0.68956743 |
| K Neighbors Classifier | 0.676470588 | 0.755725191 |
| Logistic Regression | 0.62915601 | 0.623409669 |
| SVC | 0.657289003 | 0.676844784 |

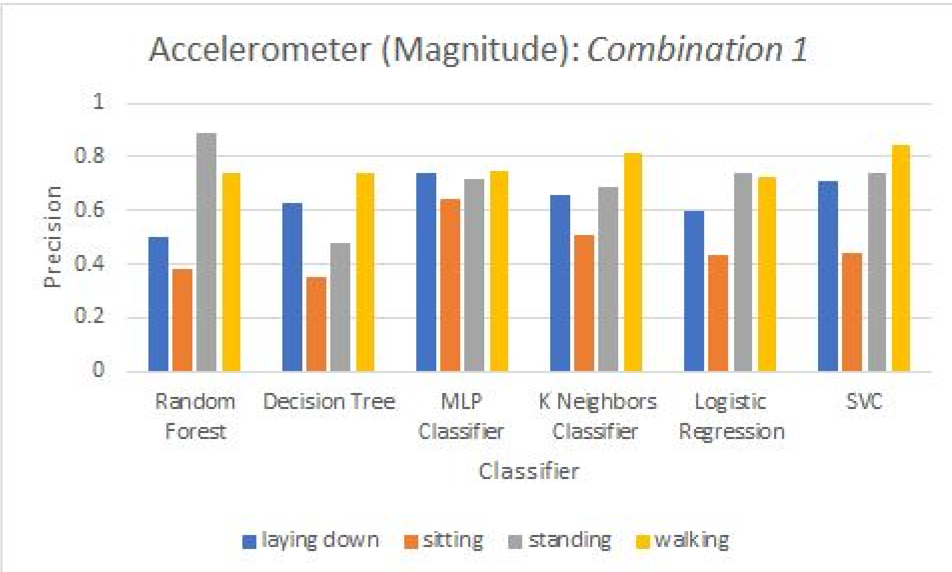



### Precision:

Combination 1

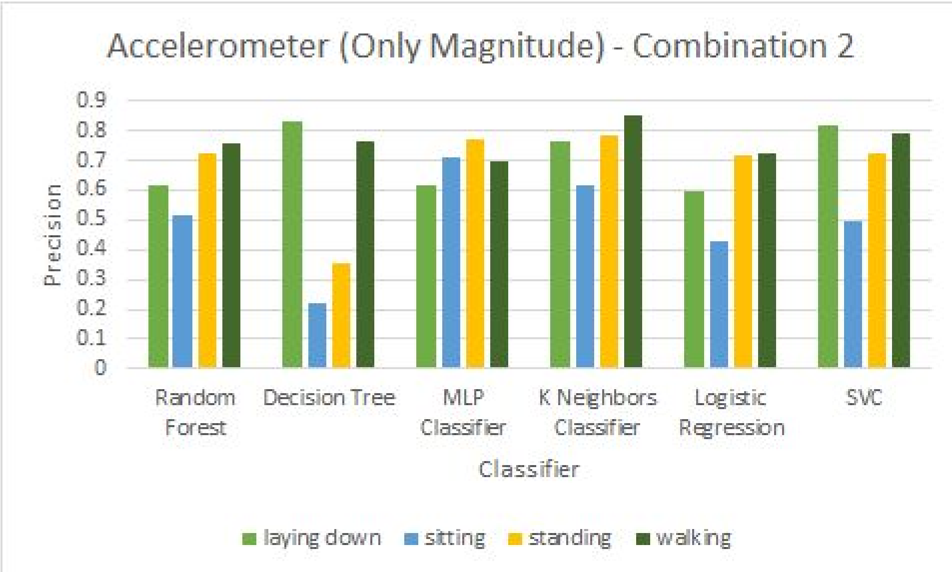
| | laying down | sitting | standing | walking |
|---|---|---|---|---|
| Random Forest | 0.50352113 | 0.38053097 | 0.88815789 | 0.74248927 |
| Decision Tree | 0.63070539 | 0.35135135 | 0.4806867 | 0.73931624 |
| MLP Classifier | 0.73958333 | 0.64397906 | 0.7202381 | 0.74891775 |

| | | | | |
|---|---|---|---|---|
| K Neighbors Classifier | 0.65693431 | 0.51162791 | 0.68461538 | 0.81220657 |
| Logistic Regression | 0.59447005 | 0.43452381 | 0.74050633 | 0.72384937 |
| SVC | 0.71232877 | 0.44404332 | 0.73617021 | 0.84263959 |

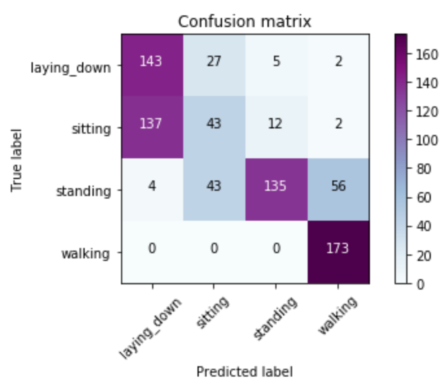
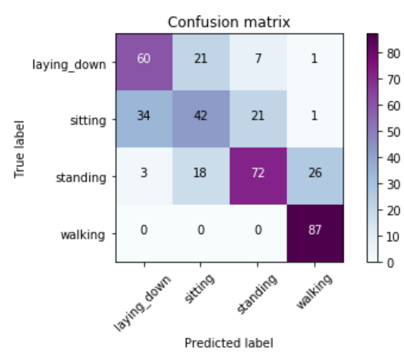
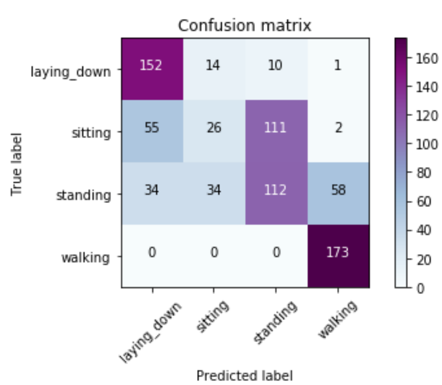
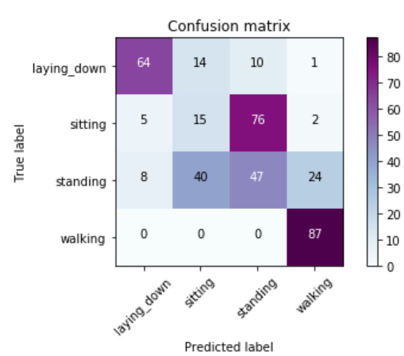



<u>Combination 2</u>

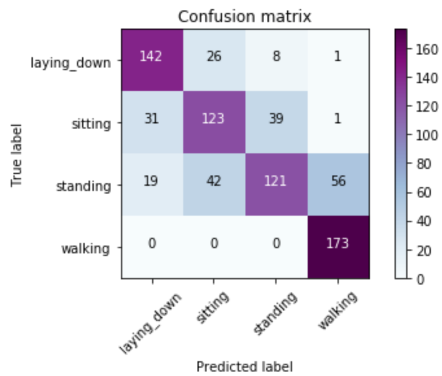
| | laying down | sitting | standing | walking |
|---|---|---|---|---|
| Random Forest | 0.6185567 | 0.51851852 | 0.72 | 0.75652174 |
| Decision Tree | 0.83116883 | 0.2173913 | 0.35338346 | 0.76315789 |
| MLP Classifier | 0.6147541 | 0.71186441 | 0.77011494 | 0.696 |
| K Neighbors Classifier | 0.765625 | 0.61386139 | 0.78571429 | 0.85294118 |
| Logistic Regression | 0.59803922 | 0.42696629 | 0.7195122 | 0.725 |
| SVC | 0.81632653 | 0.49618321 | 0.72222222 | 0.79047619 |

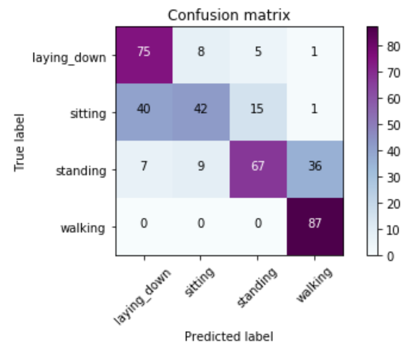



## Confusion Matrix:

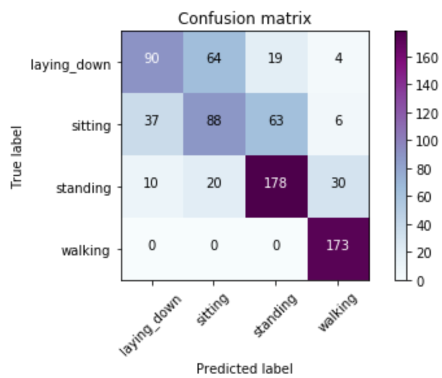
| | Combination 1 | Combination 2 |
|---|---|---|
| Random Forest |   |   |
| Decision Tree |   |   |

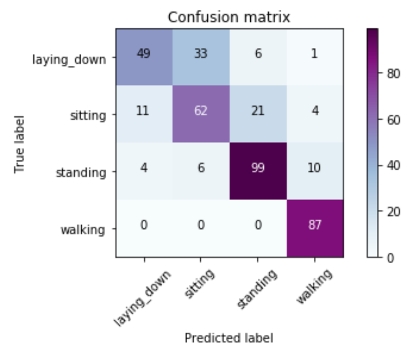
MLP Classifier



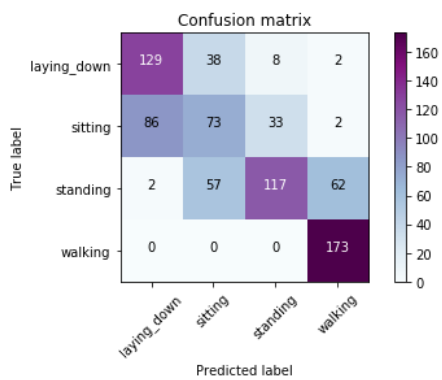



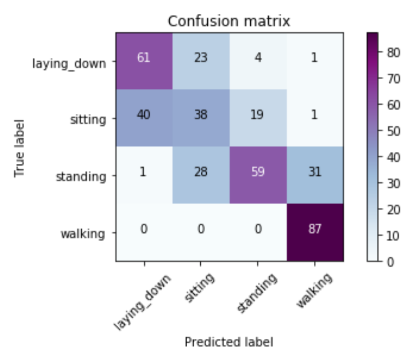

K Neighbors Classifier



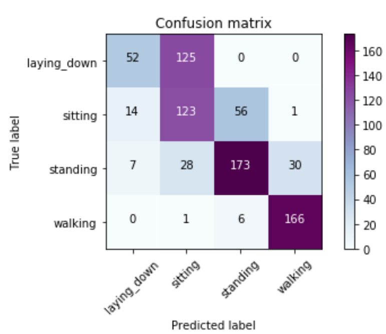



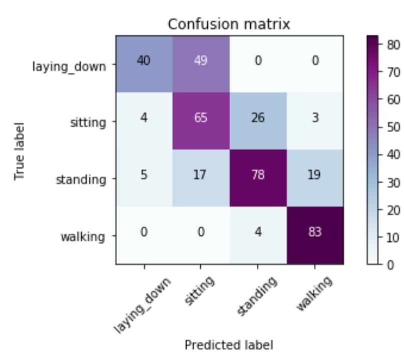

Logistic Regression




SVC



## Observations & Key Takeaways

- Initially we performed cross validation on the classification models using the entire dataset with 10 folds. The cross validation scores were very good giving consistent accuracy of over 90% across all models
- Separating out Training and Testing/Validation dataset shows a drastic reduction in accuracy with the accuracy falling to as low as 34% in some cases. This demonstrates that the results obtained from cross validation do not give a good estimate of the performance of the models
- Increase in Training data to include more people resulted in a significant increase in accuracy and hence makes our model more robust
- Hyperparameter tuning of classifiers is an integral part of creating a good model for classification
- Out of all our models, the best performing models were for the combination of accelerometer and gyroscope sensors including the magnitude features. Consistently high accuracies were achieved in the range of 0.65 to 0.82 with the highest being achieved for the K Neighbors Classifier and the lowest for Logistic Regression.
- The models trained on only magnitude features gave the lowest accuracy, but one interesting thing to note was that the classification for the walking labels was extremely accurate as can be observed from the confusion matrices.
- Of all the classifiers tested, the best results were obtained from the K Neighbors Classifiers not only in terms of accuracy but also precision. The worst performing classifier was the Logistic Regression Model.
- While the accuracy of predicting walking and laying down was high for all classifiers, it was difficult to distinguish between sitting and standing because of the similarity between the actions.
- We used both overlapping and non-overlapping windows for our analysis and we observed that the performance from both the techniques was comparable and both gave good results in the form of high accuracy and good confusion matrices.

## Challenges

We faced a number of challenges while doing this project. Some of them included:

- Data collection:
    - Ensuring the quality of data: Keeping a track of whether the data was clean

- or skewed was a little difficult. A small change in the quality of the data also resulted in a change(quite often) drop in the accuracy of the results.
  - Collecting equal amount of data for each class: Maintaining a balance in the kind of data for the 4 labels was difficult. We had to make sure that walking, sitting, standing and laying down were all equally in proportion. From all the labels ‘standing’ was the most difficult to collect since standing at a particular place for a given amount of time was not easy for the users.

- Hyperparameter tuning of classifiers:
  - Grid search takes a considerable amount of time and computing power.
  - Especially in the case of random forest, the grid search had to be terminated in about 90 mins since the code would go on running for probably a day if it wasn't stopped manually.

## Conclusion

- Various classification techniques such as Decision Trees, Support Vector Machine, Random Forests can be effectively used to classify activities of people based on raw sensor data obtained from a smartwatch.

- Data preprocessing, intelligent feature selection, and hyperparameter tuning are key components which help improve the accuracy of the classification models.

- To further boost the current results of our models, several techniques such as smoothing, filtering using band pass filter to remove noise, and incorporation of additional frequency domain features can be explored in the future.

## References

[1] Tapia, Emmanuel Munguia, Stephen S. Intille, and Kent Larson. "Activity recognition in the home using simple and ubiquitous sensors." International conference on pervasive computing. Springer, Berlin, Heidelberg, 2004.

[2] Mannini, Andrea, and Angelo Maria Sabatini. "Machine learning methods for classifying human physical activity from on-body accelerometers." Sensors 10.2 (2010): 1154-1175.

[3] Liu, Shaopeng, Robert Gao, and Patty Freedson. "Computational methods for estimating energy expenditure in human physical activities." Medicine and science in sports and exercise 44.11 (2012): 2138.

[4] Rosenberger, Mary E., et al. "Estimating activity and sedentary behavior from an accelerometer on the hip or wrist." Medicine and science in sports and exercise 45.5 (2013): 964.

[5] Mannini, Andrea, et al. "Activity recognition using a single accelerometer placed at the wrist or ankle." Medicine and science in sports and exercise 45.11 (2013): 2193.

[6] Ferscha, Alois, et al. "Orientation sensing for gesture-based interaction with smart artifacts." Computer communications 28.13 (2005): 1552-1563.

[7] White, Tom, et al. "Estimation of physical activity energy expenditure during free-living from wrist accelerometry in UK adults." PLoS One 11.12 (2016): e0167472.

[8] Mannini, Andrea, et al. "Activity Recognition in Youth Using Single Accelerometer Placed at Wrist or Ankle." Medicine and science in sports and exercise 49.4 (2017): 801-812.

[9] Erdaş, Ç. Berke, et al. "Integrating features for accelerometer-based activity recognition." Procedia Computer Science 98 (2016): 522-527.

[10] Incel, Ozlem Durmaz. "Analysis of movement, orientation and rotation-based sensing for phone placement recognition." Sensors 15.10 (2015): 25474-25506.

[11] S.-W. Lee and K. Mase. Activity and location recognition using wearable sensors. IEEE Pervasive Computing, 1(3): 24–32, 2002.

[12] J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pages 747–52. IEEE Press, 2001.

[13] J.B. Bussmann, W.L. Martens, J.H. Tulen, F.C. Schasfoort, H.J. van den Berg-Emons, and H.J. Stam. Measuring daily behavior using ambulatory accelerometry: the Activity Monitor. Behavior Research Methods, Instruments, & Computers,33(3):349–56, 2001.

[14] N. Kern, B. Schiele, and A. Schmidt. Multi-sensor activity context detection for wearable computing. In European Symposium on Ambient Intelligence (EUSAI). 2003.

[15] G. Welk and J. Differding. The utility of the Digi-Walker step counter to assess daily physical activity patterns. Medicine & Science in Sports & Exercise, 32(9):S481–S488, 2000.

[16] K. Van Laerhoven, A. Schmidt, and H.-W. Gellersen. Multi-sensor context aware clothing. In Proceedings of the 6th IEEE International Symposium on Wearable Computers, pages 49–56. IEEE Press, 2002.

[17] A. Krause, D.P. Siewiorek, A. Smailagic, and J. Farringdon. Unsupervised, dynamic identification of physiological and activity context in wearable computing. In Proceedings of the 7th International Symposium on Wearable Computers, pages 88–97. IEEE Press, 2003.

[18] F. Foerster, M. Smeja, and J. Fahrenberg. Detection of posture and motion by accelerometry: a validation in ambulatory monitoring. Computers in Human Behavior, 15:571–583, 1999.

[19] Stephen J. Preece*, John Yannis Goulermas, Member, IEEE, Laurence P. J. Kenney, and David Howard: A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 56, NO. 3, MARCH 2009