Human Activity Recognition

Contents

[Overview 2](#_Toc484970715)

[Purpose 2](#_Toc484970716)

[State of the Art 3](#_Toc484970717)

[Sensor based user activity recognition. 3](#_Toc484970718)

[Activity recognition through logic and reasoning 4](#_Toc484970719)

[Data mining based approach to activity recognition 5](#_Toc484970720)

[Vision-based activity recognition 5](#_Toc484970721)

[Data – Human Activity 6](#_Toc484970722)

[Overview 6](#_Toc484970723)

[Description 6](#_Toc484970724)

[Data Dimensions 7](#_Toc484970725)

[Data Preview 7](#_Toc484970726)

[Data visualization 8](#_Toc484970727)

[Method 9](#_Toc484970728)

[Supervised Machine learning approach 9](#_Toc484970729)

[Data Pre-processing 9](#_Toc484970730)

[Principal Component Analysis (Dimensionality reduction) 9](#_Toc484970731)

[Decision Tree Classification 10](#_Toc484970732)

[Random Forest Classification 10](#_Toc484970733)

[Support Vector Machine 10](#_Toc484970734)

[*k*-nearest neighbors 11](#_Toc484970735)

[Results 11](#_Toc484970736)

[Data Pre-Processing 11](#_Toc484970737)

[Principal Component Analysis 12](#_Toc484970738)

[Variance explained by components 12](#_Toc484970739)

[PCA Visualization – Comp1 Vs Comp2 13](#_Toc484970740)

[Decision tree model on PCA data 13](#_Toc484970741)

[SVM model on PCA data 14](#_Toc484970742)

[kNN on PCA data 15](#_Toc484970743)

[kNN on actual data 16](#_Toc484970744)

[Decision tree model on actual data 16](#_Toc484970745)

[SVM model on actual data 18](#_Toc484970746)

[Random Forest model on actual data (ntree=100) 19](#_Toc484970747)

[Random Forest tuned (ntree=200, mtry=12) 20](#_Toc484970748)

[kNN on actual data 21](#_Toc484970749)

[Ensemble - Stacking 21](#_Toc484970750)

[Ensemble - SVM 21](#_Toc484970751)

[Analysis 22](#_Toc484970752)

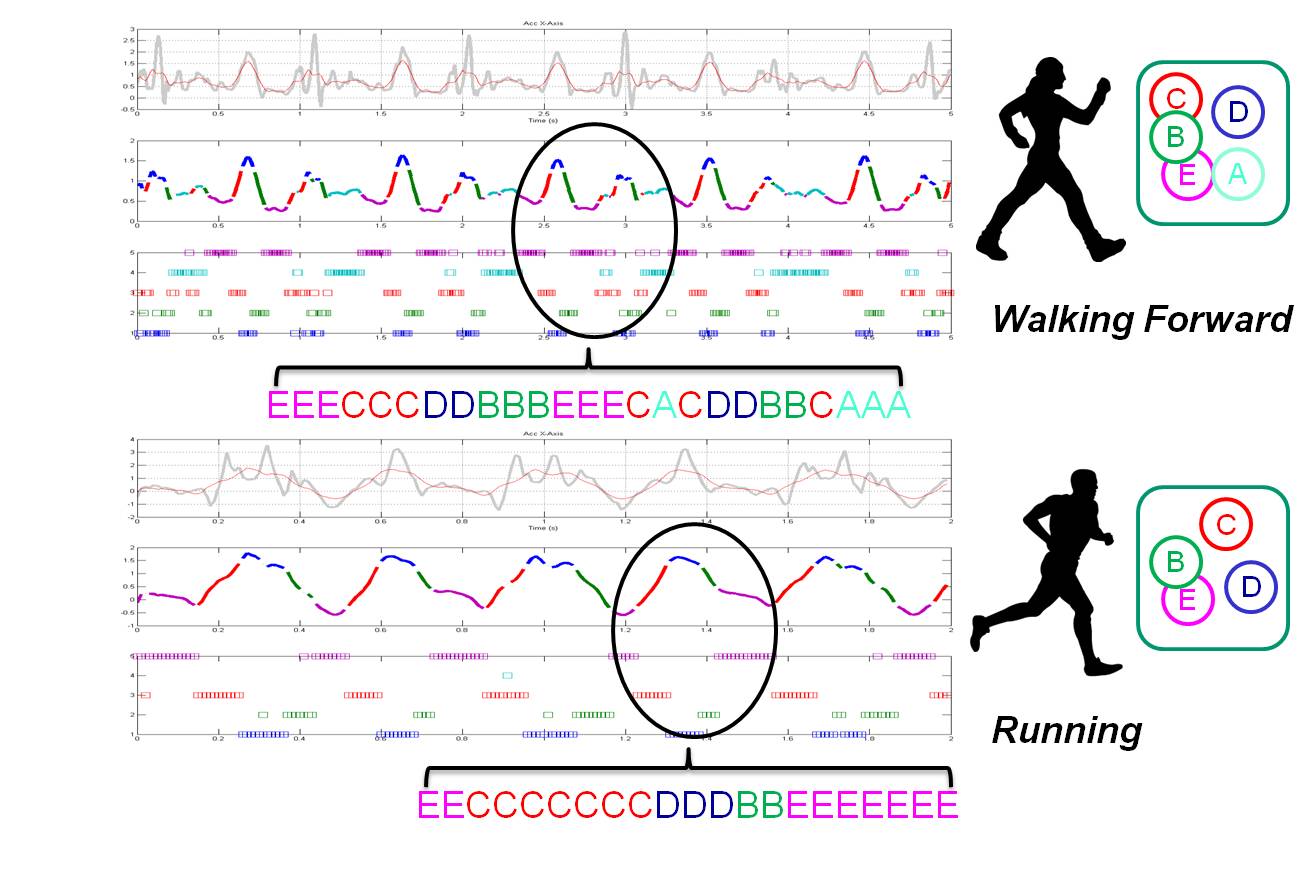
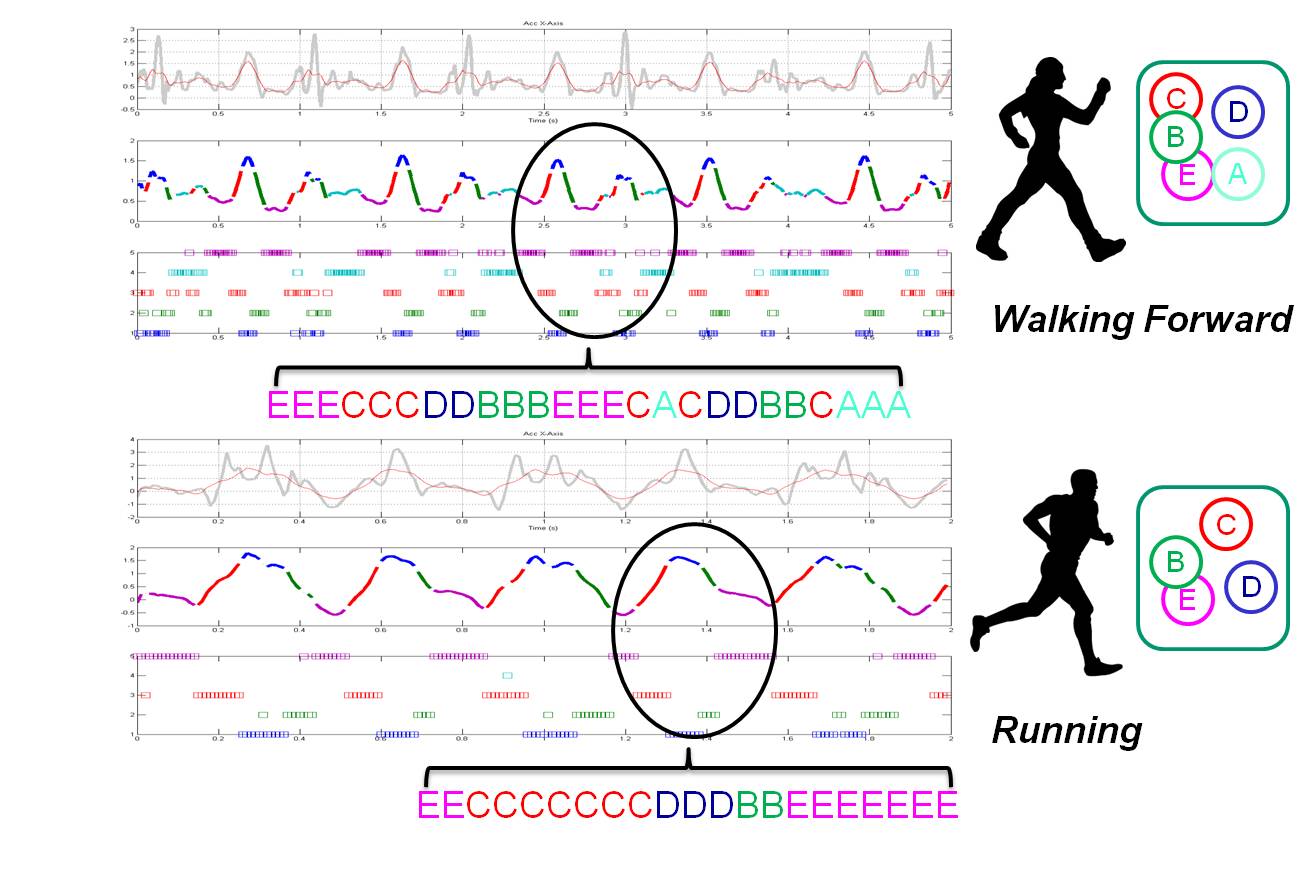
# Overview

Human activity recognition has seen a tremendous growth in the last decade playing a major role in the field of health care and research. This emerging popularity can be attributed to its many of real-life applications primarily dealing with human-centric problems like healthcare and eldercare. Many research attempts with data mining and machine learning techniques have been undergoing to accurately detect human activities for e-health systems.

# Purpose

To understand activity recognition better, consider the following elderly assistance scenario. An elderly man wakes up at dawn in his apartment, where he stays alone. He lights the stove to make a pot of tea, switches on the toaster oven, and takes some bread and jelly from the cupboard. After he takes his morning medication, a computer-generated voice gently reminds him to turn off the toaster. Later that day, his daughter accesses a secure website where she scans a check-list, which was created by a sensor network in her father's apartment. She finds that her father is eating normally, taking his medicine on schedule, and continuing to manage his daily life on his own.

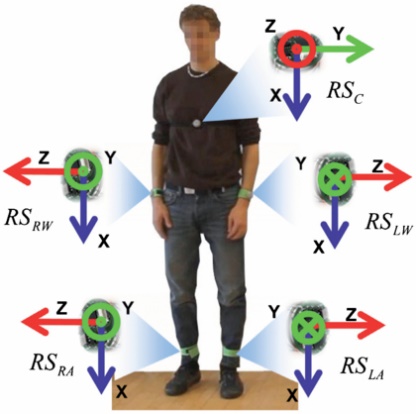
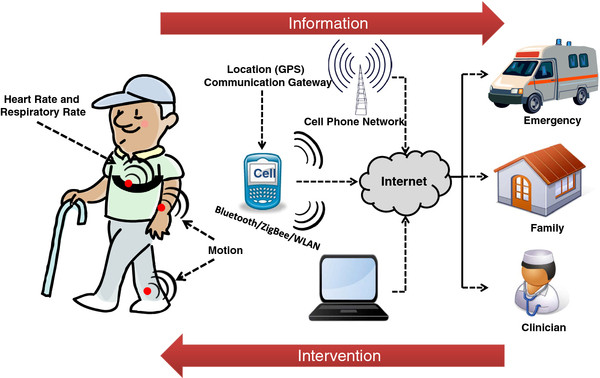
Due to its many-faceted nature, different fields may refer to activity recognition as plan recognition, goal recognition, intent recognition, behaviour recognition, location estimation and location-based services.



# State of the Art

## Sensor based user activity recognition.

[Sensor](https://en.wikipedia.org/wiki/Sensor)-based activity recognition integrates the emerging area of sensor networks with novel [data mining](https://en.wikipedia.org/wiki/Data_mining) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning) techniques to model a wide range of human activities. Mobile devices (e.g. smart phones) provide sufficient sensor data and calculation power to enable physical activity recognition to provide an estimation of the energy consumption during everyday life. This process can be extended to multi user and group activity recognition.



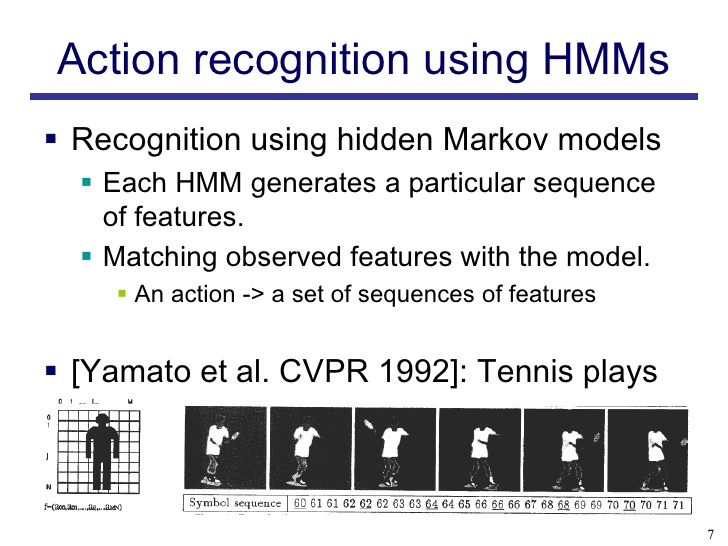
## Activity recognition through logic and reasoning

Probability theory and statistical learning models are more recently applied in activity recognition to reason about actions, plans and goals under uncertainty. In the literature, there have been several approaches which explicitly represent uncertainty in reasoning about an agent's plans and goals.

Using sensor data as input, Hodges and Pollack designed machine learning-based systems for identifying individuals as they perform routine daily activities such as making coffee. [Intel Research (Seattle) Lab](https://en.wikipedia.org/wiki/Intel_Research_Lablets) and [University of Washington](https://en.wikipedia.org/wiki/University_of_Washington) at Seattle have done some important works on using sensors to detect human plans. Some of these works infer user transportation modes from readings of radio-frequency identifiers (RFID) and global positioning systems (GPS).

The use of temporal probabilistic models has been shown to perform well in activity recognition and generally outperform non-temporal models. Generative models such as the Hidden Markov Model (HMM) and the more generally formulated Dynamic Bayesian Networks (DBN) are popular choices in modelling activities from sensor data. Discriminative models such as Conditional Random Fields (CRF) are also commonly applied and also give good performance in activity recognition.

Conventional temporal probabilistic models such as the hidden Markov model (HMM) and conditional random fields (CRF) model directly model the correlations between the activities and the observed sensor data. In recent years, increasing evidence has supported the use of hierarchical models which take into account the rich hierarchical structure that exists in human behavioural data. The core idea here is that the model does not directly correlate the activities with the sensor data, but instead breaks the activity into sub-activities (sometimes referred to as actions) and models the underlying correlations accordingly. An example could be the activity of preparing spaghetti, which can be broken down into the sub-activities or actions of cutting vegetables, frying the vegetables in a pan and serving it on a plate. Examples of such a hierarchical model are Layered Hidden Markov Models (LHMMs)[[27]](https://en.wikipedia.org/wiki/Activity_recognition#cite_note-:1-27) and the hierarchical hidden Markov model (HHMM), which have been shown to significantly outperform its non-hierarchical counterpart in activity recognition.



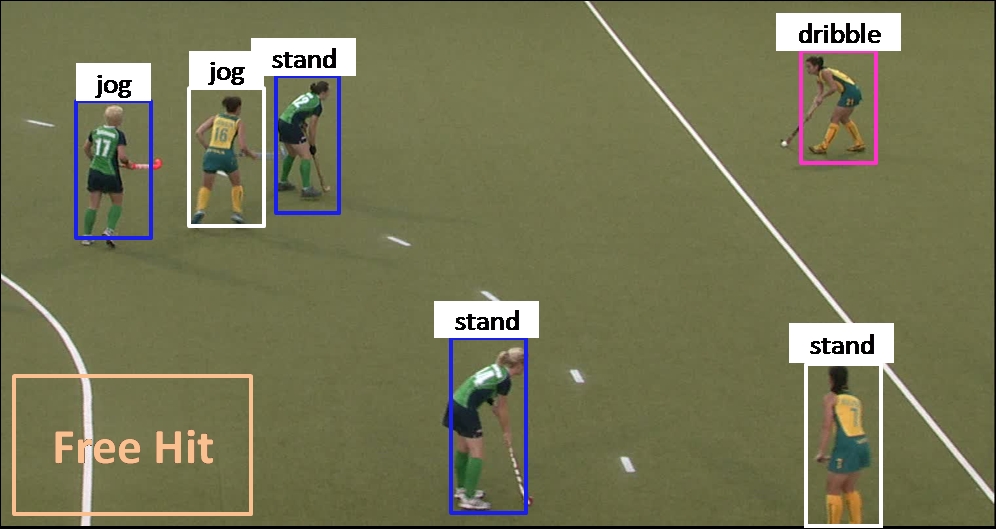
## Data mining based approach to activity recognition

Different from traditional machine learning approaches, an approach based on data mining has been recently proposed. The problem of activity recognition is formulated as a pattern-based classification problem. They proposed a data mining approach based on discriminative patterns which describe significant changes between any two activity classes of data to recognize sequential, interleaved and concurrent activities in a unified solution with use of 2D corners in both space and time. These are grouped spatially and temporally using a hierarchical process, with an increasing search area. At each stage of the hierarchy, the most distinctive and descriptive features are learned efficiently through data mining (Apriori rule).

## Vision-based activity recognition

It is a very important and challenging problem to track and understand the behaviour of agents through videos taken by various cameras. The primary technique employed is computer vision. Vision-based activity recognition has found many applications such as human-computer interaction, user interface design, robot learning, and surveillance, among others. Scientific conferences where vision based activity recognition work often appears are [ICCV](https://en.wikipedia.org/wiki/ICCV) and [CVPR](https://en.wikipedia.org/wiki/CVPR).

In vision-based activity recognition, a great deal of work has been done. Researchers have attempted a number of methods such as [optical flow](https://en.wikipedia.org/wiki/Optical_flow), [Kalman filtering](https://en.wikipedia.org/wiki/Kalman_filtering), [Hidden Markov models](https://en.wikipedia.org/wiki/Hidden_Markov_model), etc., under different modalities such as single camera, stereo, and infrared. In addition, researchers have considered multiple aspects on this topic, including single pedestrian tracking, group tracking, and detecting dropped objects.

Recently some researchers have used RGBD cameras like Microsoft Kinect to detect human activities. Depth cameras add extra dimension i.e. depth which normal 2d camera fails to provide. Sensory information from these depth cameras have been used to generate real-time skeleton model of humans with different body positions. These skeleton information provides meaningful information that researchers have used to model human activities which are trained and later used to recognize unknown activities

# Data – Human Activity

### Overview

The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The objective is to classify activities into one of the six activities performed.

### Description

The experiments have been carried out with a **group of 30 volunteers** within an age bracket of 19-48 years. Each person performed **six activities** (WALKING, WALKING\_UPSTAIRS, WALKING\_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a **smartphone** (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

The sensor signals (accelerometer and gyroscope) were **pre-processed** by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

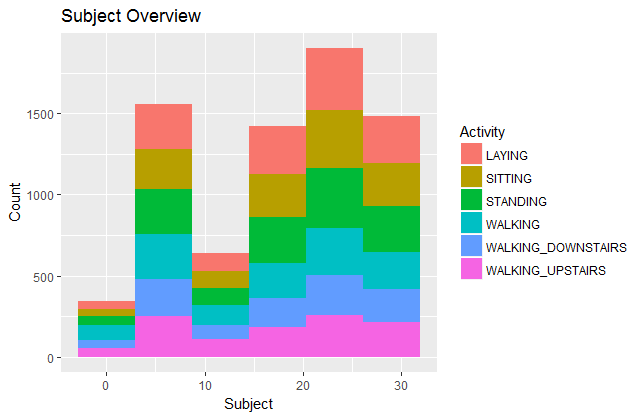
### Data Dimensions

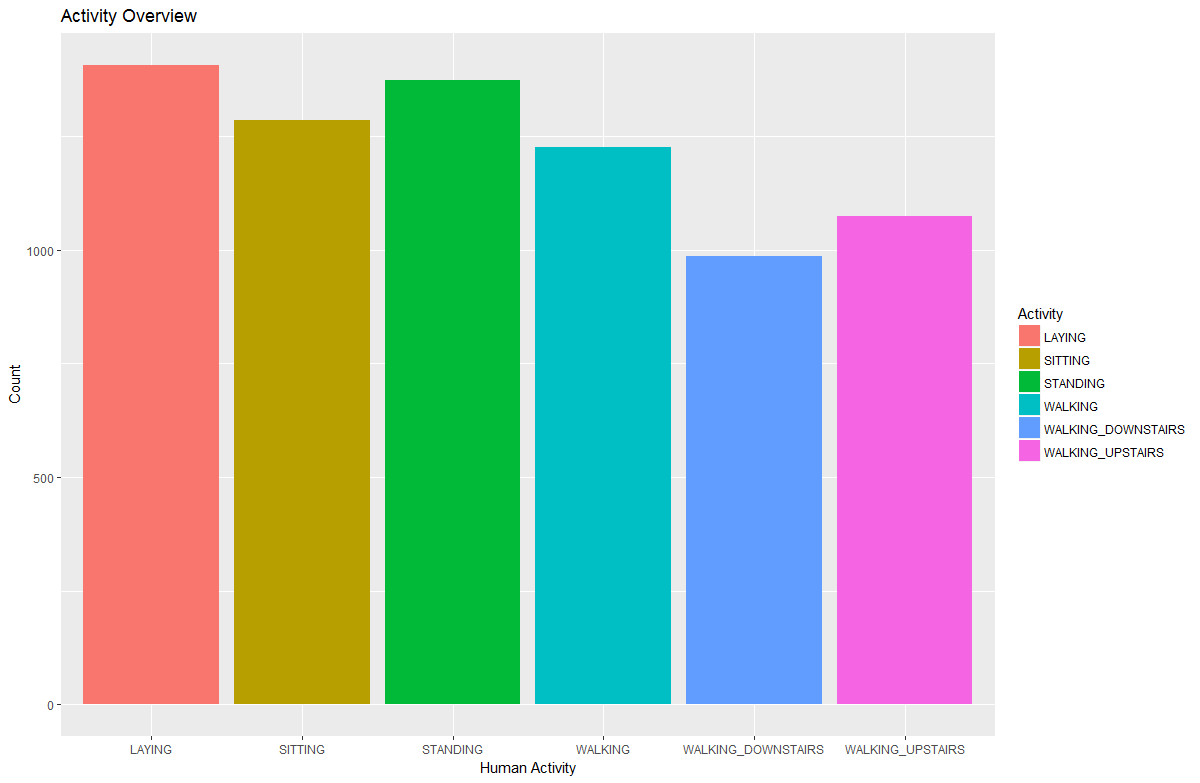
|  |  |  |
| --- | --- | --- |
| Data | #Rows | #Columns |
| Train | 7352 | 563 |
| Test | 2947 | 563 |

### Data Preview



### Data visualization





# Method

## Supervised Machine learning approach

Feature engineering technique such as PCA can be applied to reduce dimensions of the data and make the data for model friendly. Classification techniques such as Decision Tree, Random Forest, SVM and KNN gives better results on the given data / PCA data. Finally ensemble technique such as stacking can be applied to get the best accuracy.

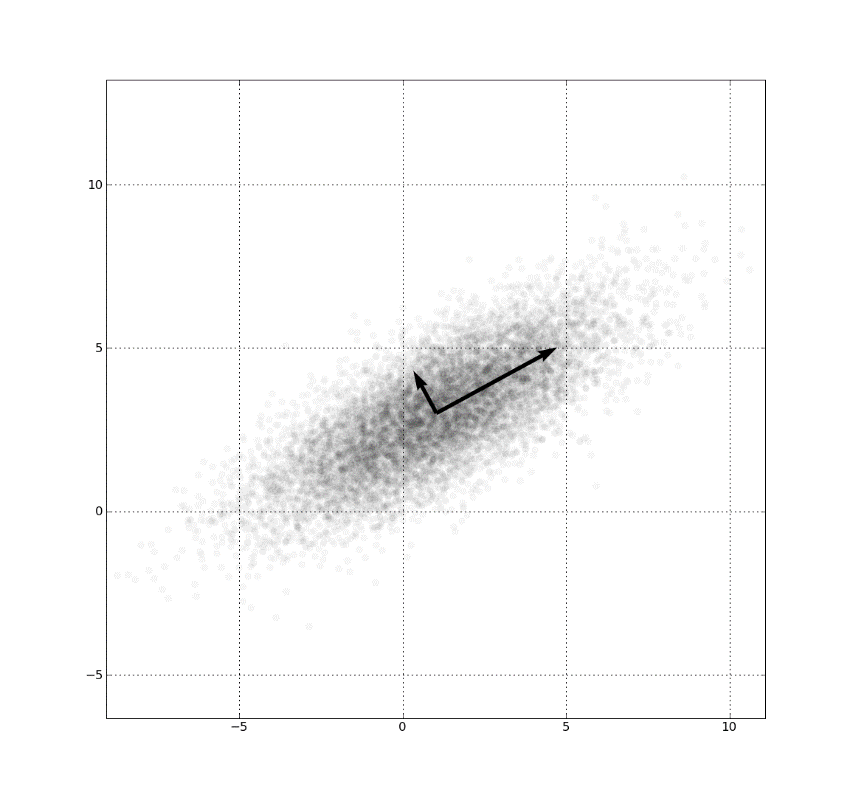
### Data Pre-processing

Pre-processing data is very important task to perform in order to make the data model friendly and to obtain maximum model accuracy. Below are the pre-processing approach.

* Identifying the missing values
* Impute missed values if any
* Binning
* Feature Engineering
* Feature reduction
* Normalizing data
* Splitting the train data into train,validate

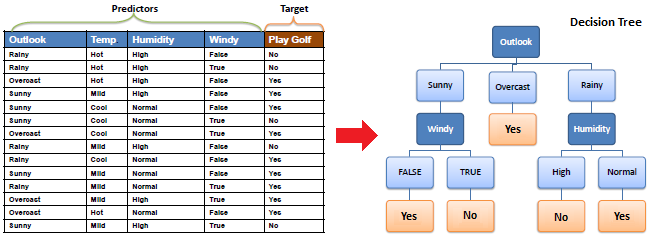
### Principal Component Analysis (Dimensionality reduction)

A method of analysis which involves finding the linear combination of a set of variables that has maximum variance and removing its effect, repeating this successively.



### Decision Tree Classification

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

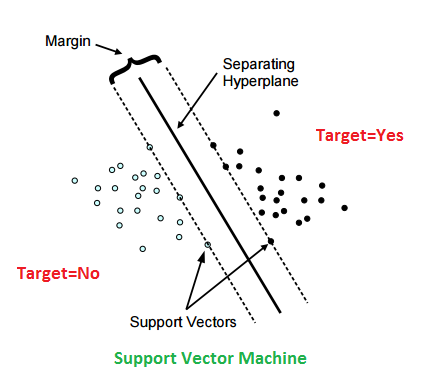


### Random Forest Classification

Random forests or random decision forests are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks, that operate by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their training set.

### Support Vector Machine

SVMs are [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) models with associated learning [algorithms](https://en.wikipedia.org/wiki/Algorithm) that analyse data used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis). Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-[probabilistic](https://en.wikipedia.org/wiki/Probabilistic_classification) [binary](https://en.wikipedia.org/wiki/Binary_classifier). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.



### *k*-nearest neighbors

k-NN is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis). In both cases, the input consists of the k closest training examples in the [feature space](https://en.wikipedia.org/wiki/Feature_space). The output depends on whether k-NN is used for classification or regression.

k-NN is a type of [instance-based learning](https://en.wikipedia.org/wiki/Instance-based_learning), or [lazy learning](https://en.wikipedia.org/wiki/Lazy_learning), where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all [machine learning](https://en.wikipedia.org/wiki/Machine_learning) algorithms.

# Results

### Data Pre-Processing

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width.

### Principal Component Analysis

Performed PCA to reduce dimensions and identified the components which are explaining the maximum variance.

Importance of components:

Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7

Standard deviation 2.9843663 0.85579435 0.78532360 0.5190205 0.49002998 0.42510505 0.41129282

Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14

Standard deviation 0.3964561 0.377133254 0.350751599 0.347924820 0.31934222 0.311364737 0.294411605

Comp.15 Comp.16 Comp.17 Comp.18 Comp.19 Comp.20

Standard deviation 0.279765272 0.265916889 0.261759390 0.256800957 0.247620669 0.243872550

Comp.21 Comp.22 Comp.23 Comp.24 Comp.25 Comp.26

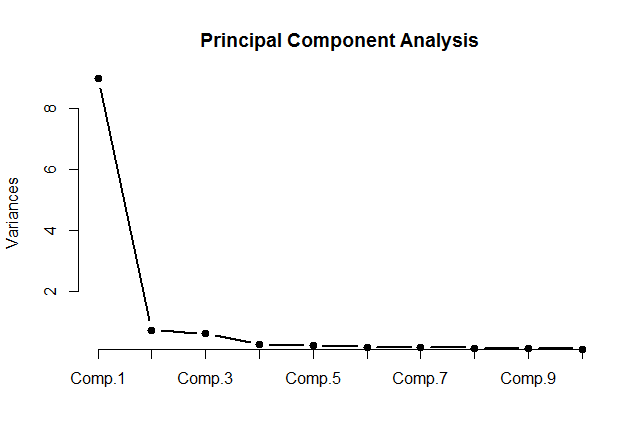
Standard deviation 0.243285551 0.236548510 0.231513515 0.226020783 0.222313718 0.216782008

Comp.27 Comp.28 Comp.29 Comp.30 Comp.31 Comp.32

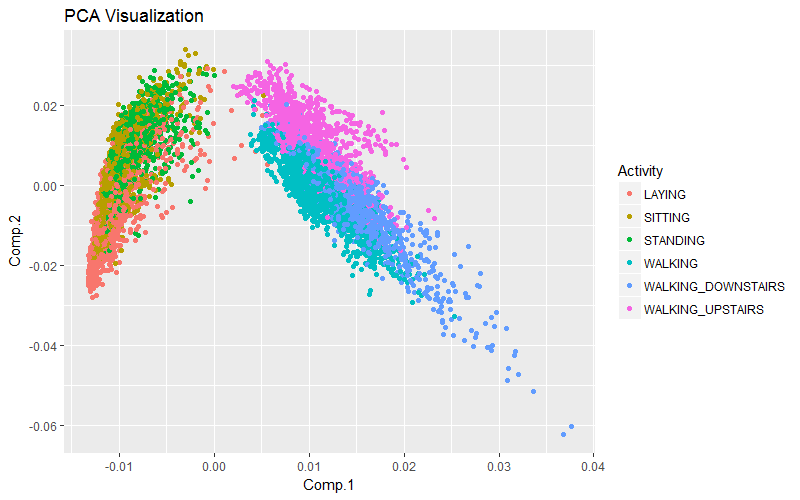
Standard deviation 0.212064108 0.209811870 0.207693838 0.204276296 0.201375711 0.197795700

Comp.33 Comp.34 Comp.35 Comp.36 Comp.37 Comp.38

### Variance explained by components



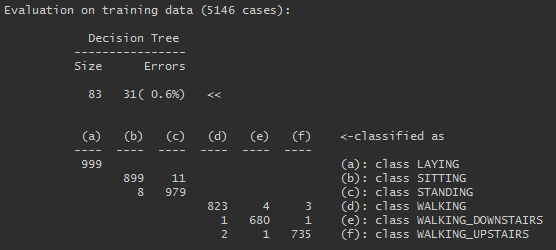
### PCA Visualization – Comp1 Vs Comp2



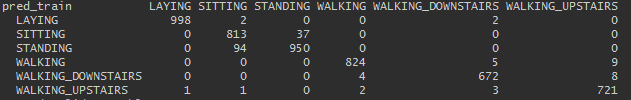
### Decision tree model on PCA data

Built decision tree model on the PCA data and checked the accuracy on Train, Validate and Test data.

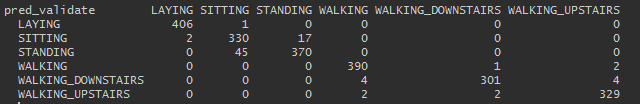
#### Summary Decision Tree



#### Prediction on train data



#### Prediction on validate data



#### Decision tree accuracy

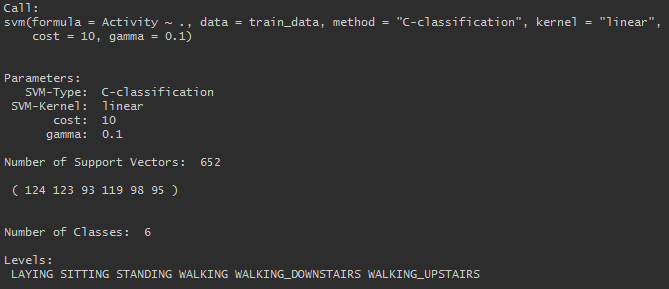
Accuracy on Train data - 96.73533

Accuracy on validate data - 96.37353

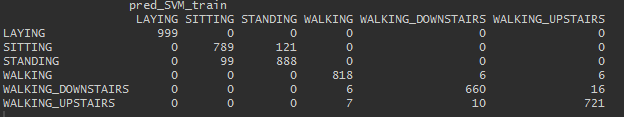
Accuracy on test data - **73.80387**

### SVM model on PCA data

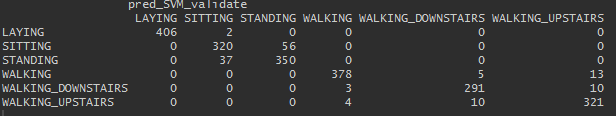
#### Summary SVM model



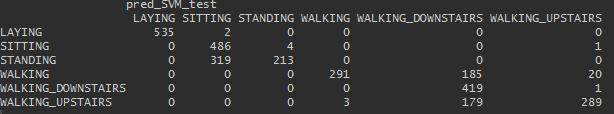
#### Prediction on Train data



#### Prediction on validate data



#### Prediction on test data



#### SVM model accuracy

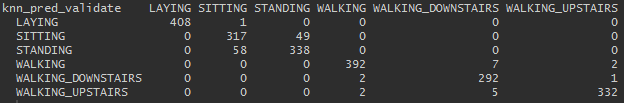
Accuracy on Train data - 94.73

Accuracy on validate data - 93.65

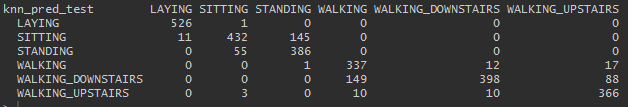
Accuracy on test data - **75.77**

### kNN on PCA data

#### Prediction on validate data



#### Prediction on test data



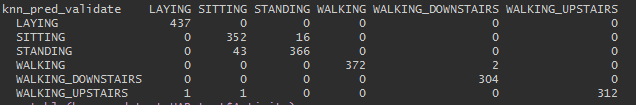
#### kNN accuracy

Accuracy on validate data - 94.24297

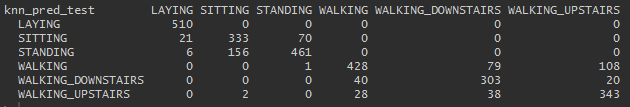
Accuracy on test data - **82.96573**

### kNN on actual data

#### Prediction on validate data



#### Prediction on test data



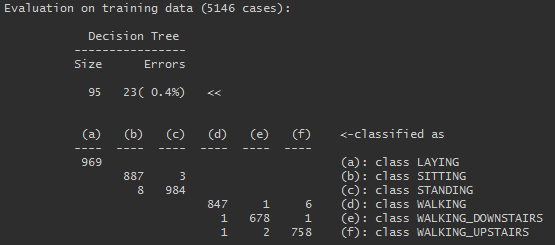
#### kNN accuracy

Accuracy on validate data - 97.14415

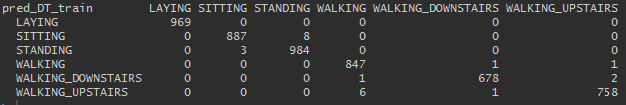
Accuracy on test data - **80.69223**

### Decision tree model on actual data

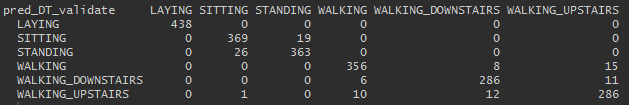
#### Summary decision tree



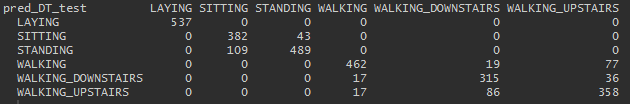
#### Prediction on train data



#### Prediction on validate data



#### Prediction on test data



#### Decision tree accuracy

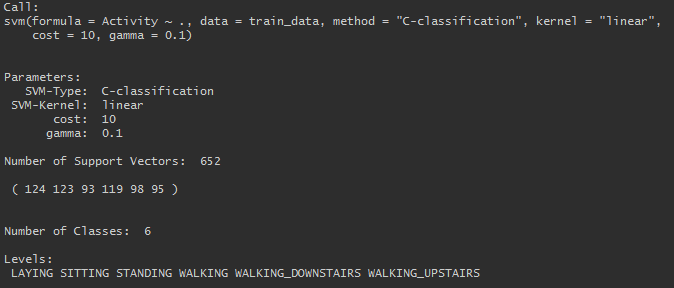
Accuracy on train data - 99.55305

Accuracy on validate data - 95.10426

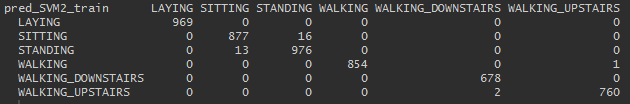
Accuracy on test data - **86.29114**

### SVM model on actual data

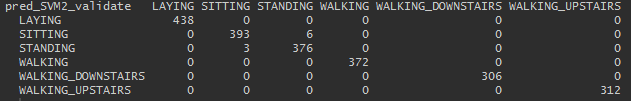
#### Summary SVM



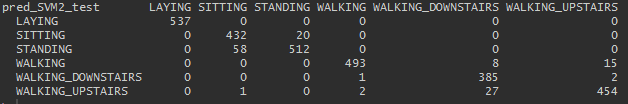
#### Prediction on train data



#### Prediction on validate data



#### Prediction on test data



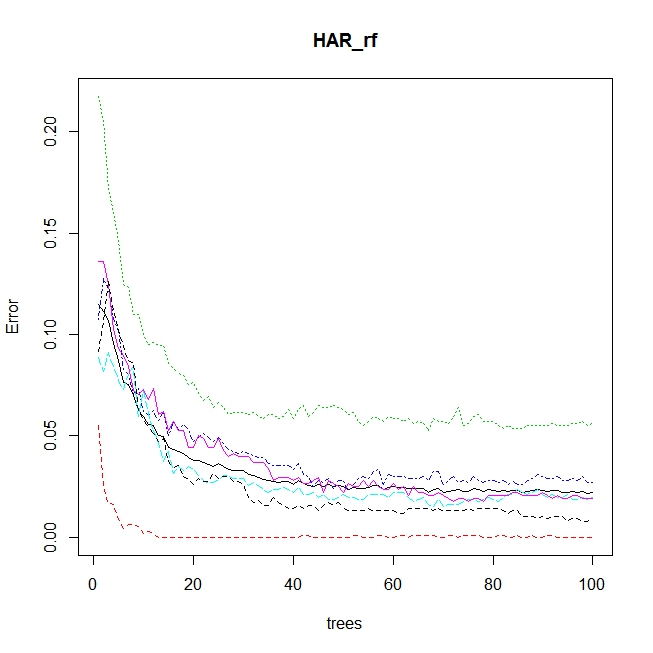
#### SVM accuracy

Accuracy on train data - 99.37816

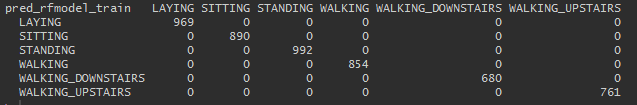
Accuracy on validate data - 99.59202

Accuracy on test data - 95.453

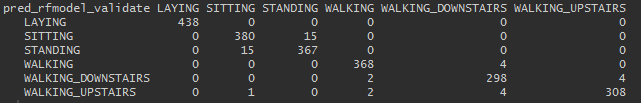
### Random Forest model on actual data (ntree=100)



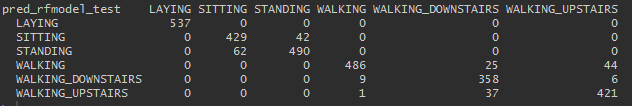
#### Prediction on train data



#### Prediction on validate data



#### Prediction on test data



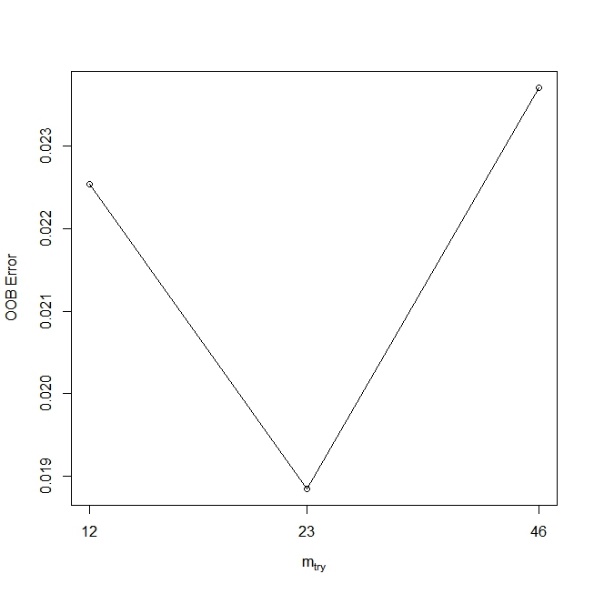
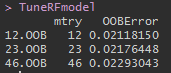
#### Random Forest accuracy

Accuracy on train data - 100

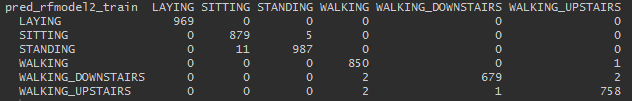
Accuracy on validate data - 97.86945

Accuracy on test data - 92.33118

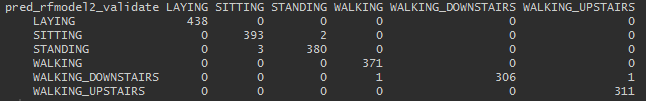
### Random Forest tuned (ntree=200, mtry=12)



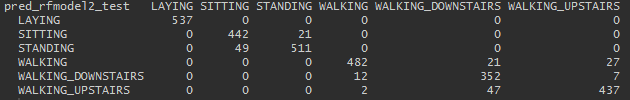
#### Prediction on train data



#### Prediction on validate data



#### Prediction on test data



#### Random Forest tuned accuracy

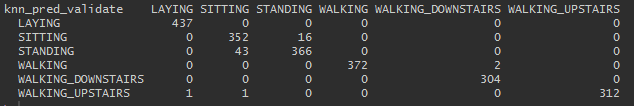
Accuracy on train data - 99.53362

Accuracy on validate data - 99.68268

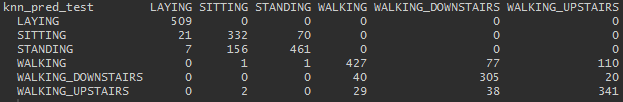
Accuracy on test data - 93.6885

### kNN on actual data

#### Prediction on validate data



#### Prediction on test data



#### kNN accuracy

Accuracy on validate data - 97.14415

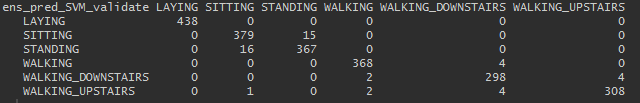
Accuracy on test data - **80.59043**

### Ensemble - Stacking

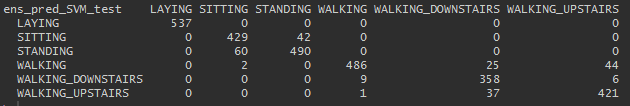
Process involves combining all the above build models predictions (Train, Validate and test), create a new dataset and build another classification model. Built SVM model for stacking as It is giving good results among all the models build on this data.

### Ensemble - SVM

#### Prediction on validate data



#### Prediction on test data



#### Ensemble SVM accuracy

Accuracy on validate data - 97.82412

Accuracy on test data - **92.33118**

# Analysis

Observed that the actual data is giving best result on SVM model than the PCA models and ensemble.