Machine learning algorithm for classification of Activity of daily life’s

Siddharth Chakravarty

Overview

ADL- What is it?

How is it monitored?

Technology landscape

Problem statement

Approach

Results

ADL- Activity of Daily life’s

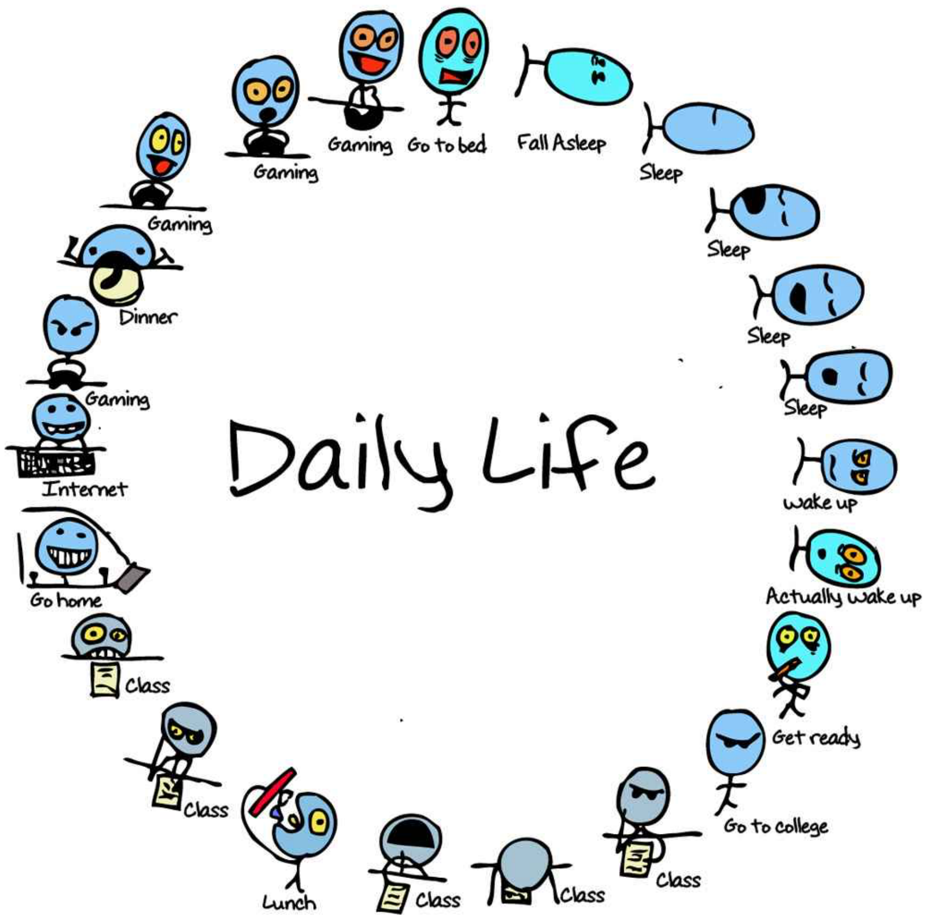
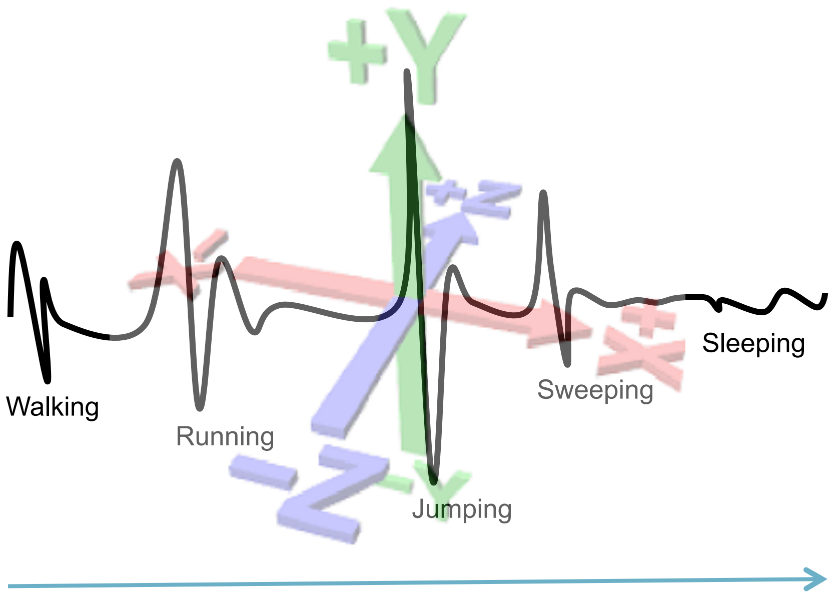


Image Source: connectingcleveland.net/wp-content/uploads/2014/07/Daily_Life_Wallpaper.jpg

A technique for classification of human activities can be a useful tool to not only classify and monitor our activities, but also improve overall quality of life

State of the art technology ADL monitoring

Image courtesy: Jawbone



- An accelerometer essentially records the acceleration it experiences is the X,Y and Z direction.
- Accelerometers are the most commonly used type of sensor for activity recognition with wearable sensors and other consumer electronic devices ranging from Iphones to Wii

Technology landscape

	Heart rate	Walking	Running	Sleeping	ADL
Heart rate monitors	X				
Pedometers	X	X	X		
Phone		X	X		
Wearable	X	X	X	X	

Major players

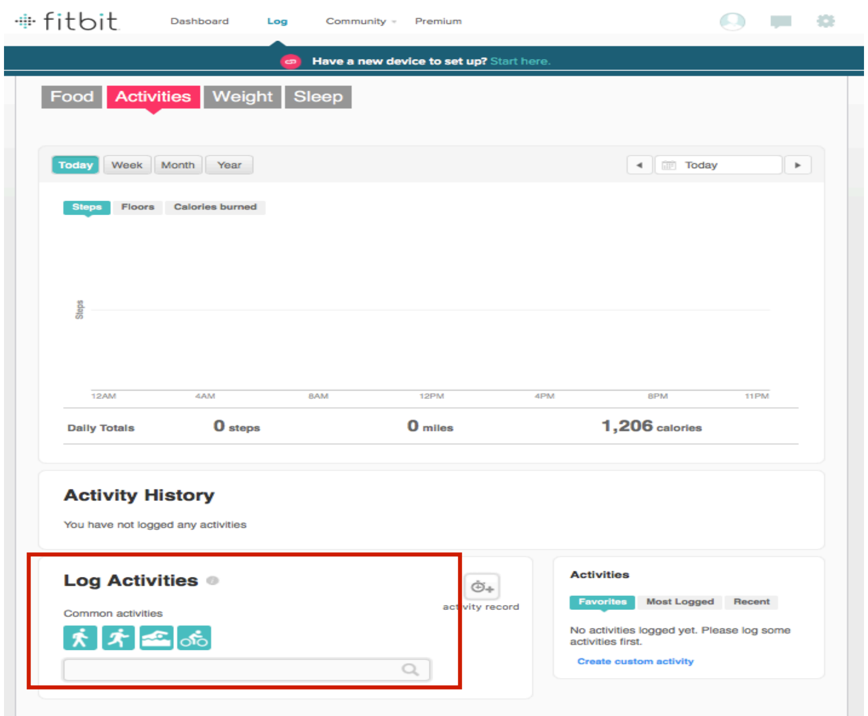
- Fitbit - recently announced next generation device
- Jawbone- acquired bodymedia to complement it's technology
- Microsoft
- Samsung
- Pebble
- Misfit– Acquired by Fossil
- Facebook- acquired Finland-based fitness app maker Protogeo

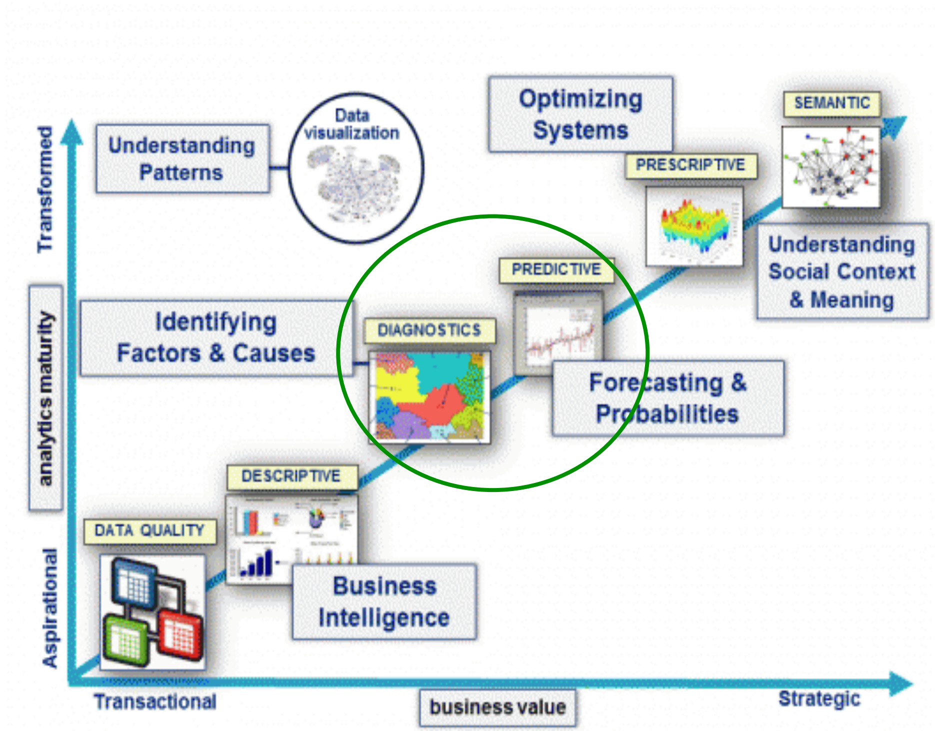
Many more….

Clearly there is a need for accurately monitor and classify ADL's not just for recreation, but also for other applications.

Problem statement

The automatic recognition of a set of Activities of Daily Living, is among the most challenging research fields in Ambient Intelligence. Main challenge with wearable technology is classification of use case ADL'

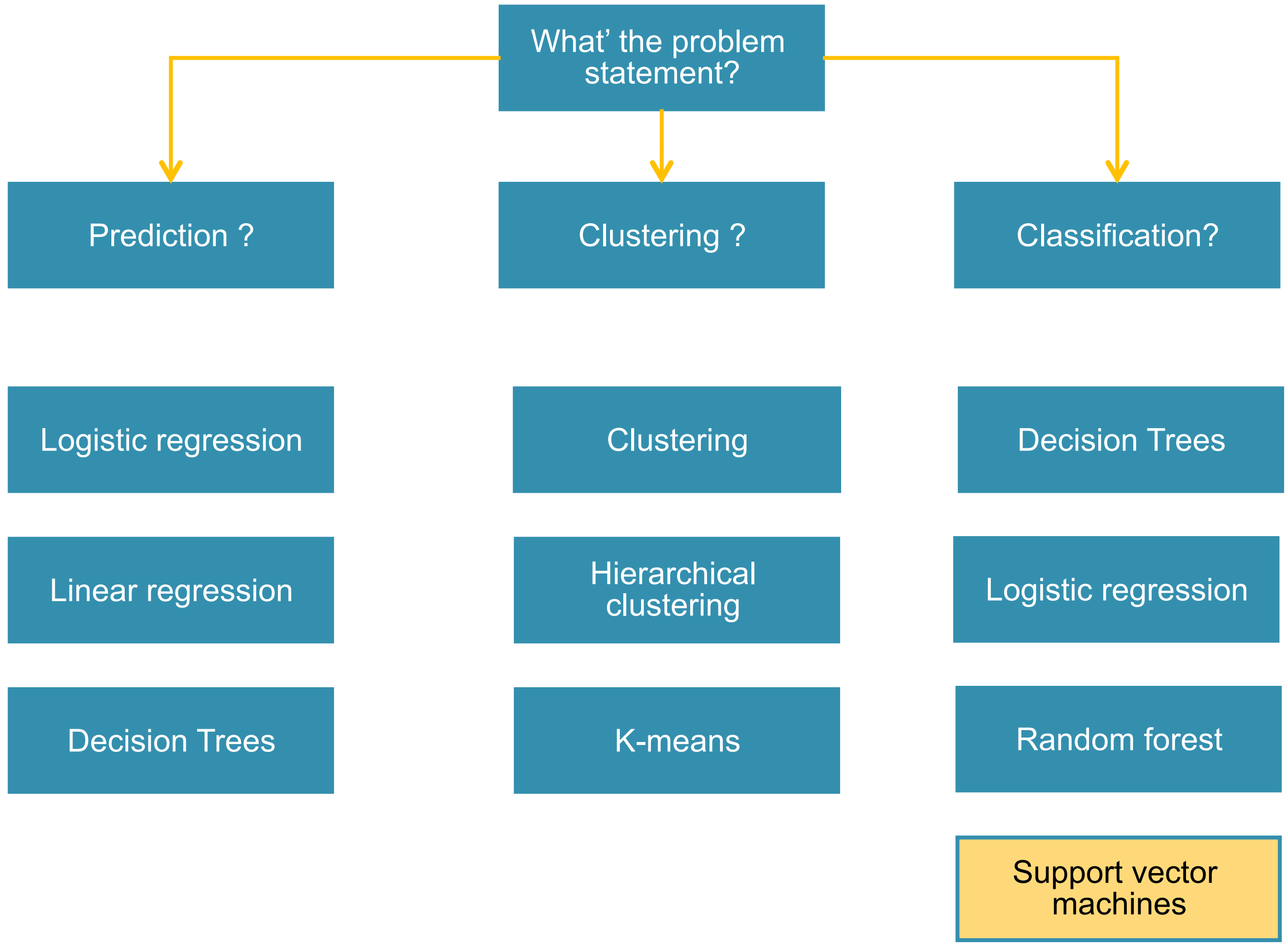




Emerging Trends in Data Analytics
Image courtesy SARK7

The objective of the project was to develop machine learning algorithm using SVM to predict ADL'.

Data science toolkit



Support vector machines SVM is a most popular and efficient classification and regression method. Currently four R packages contain SVM related software. For this project the e1071 R package was chosen that supports multi-level classification problems

The Dataset

Problem statement → $Data → Data Wrangling → Data Exploration → Build Models → Testing and classification

- The Dataset for ADL Recognition with Wrist-worn Accelerometer is a public collection of labeled accelerometer data recordings to be used for the creation and validation of acceleration models of simple ADL .
- It was provided by UCI machine learning repository, Center for Machine learning and Intelligent systems.
- The data was collected by using single tri-axial accelerometer attached to the right-wrist of the volunteer. It was carried out by Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa from the Laboratory for Ambient Intelligence and Mobile Robotics ,DIBRIS, University of Genova
- The Dataset composed of the recordings of 11 simple ADL performed by a total of 16 volunteers.

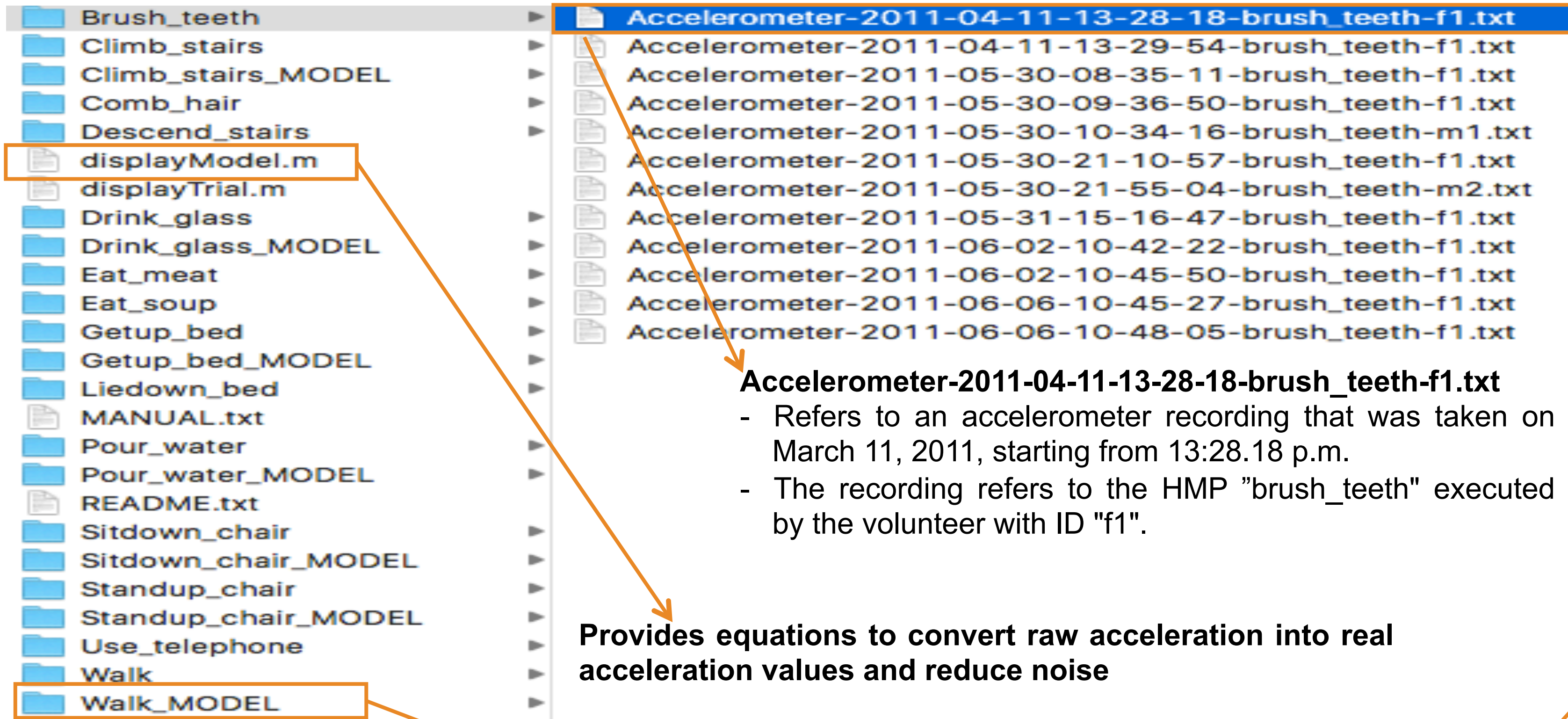
Gender		**Age**			**Weight**		
M	F	Min	Avg.	Max	Min	Avg.	Max
11	5	19	81	57.4	56	85	72.7

brush_teeth	getup_bed	walk
climb_stairs	liedown_bed	
comb_hair	pour_water	
descend_stairs	sitdown_chair	
drink_glass	standup_chair	

Wrist-worn Accelerometer
Image courtesy: Chalkbeat Colorado

$Dataset for ADL Recognition with Wrist-worn Accelerometer

Overview of dataset



Accelerometer-2011-04-11-13-28-18-brush_teeth-f1.txt

- Refers to an accelerometer recording that was taken on March 11, 2011, starting from 13:28.18 p.m.
- The recording refers to the HMP ”brush_teeth" executed by the volunteer with ID "f1".

Provides equations to convert raw acceleration into real acceleration values and reduce noise

Each folder contains raw data for the activity.

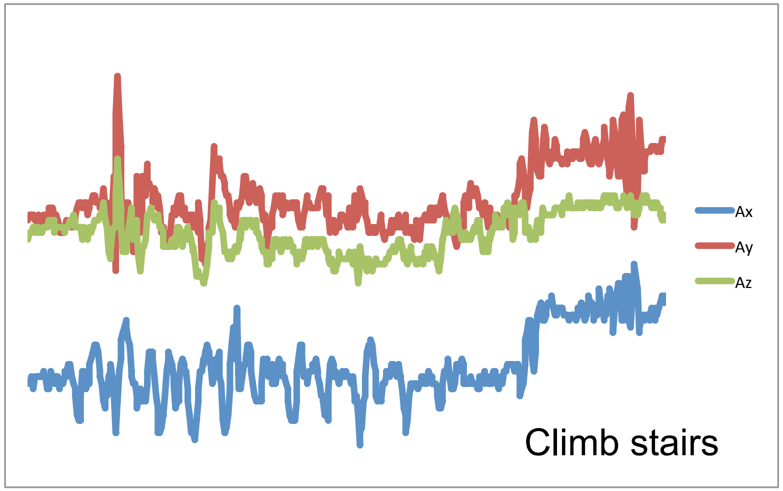
Each file in the dataset has the following naming convention: Accelerometer-[START_TIME]-[ADL]-[VOLUNTEER]

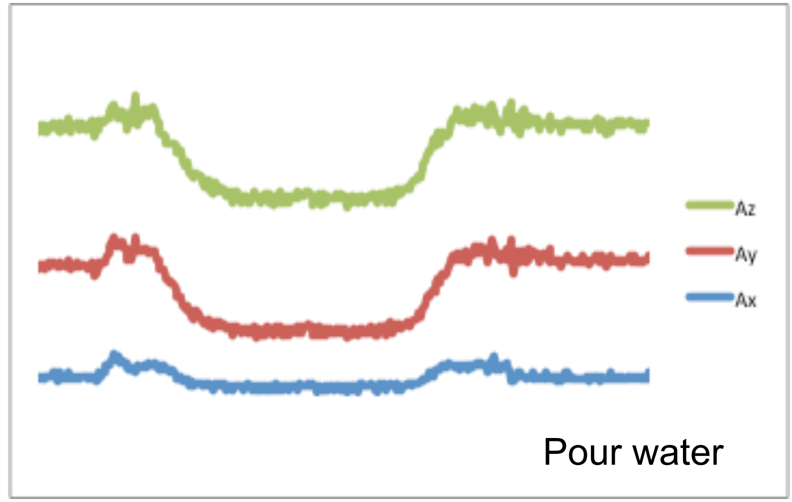
22	49	35
22	49	35
22	52	35
22	52	35
21	52	34
22	51	34
20	50	35
22	52	34
22	50	34
22	51	35
21	51	33
20	50	34
21	49	33
21	49	33
20	51	35
18	49	34
19	48	34
16	53	34
18	52	35
18	51	32
19	50	33
19	53	33
21	50	33
21	51	33
24	51	34
25	50	35
25	51	36
25	50	38
25	51	39
26	49	40
25	49	41
25	47	42
22	45	41
21	48	41
21	48	43
21	47	41
22	47	41
21	47	42
22	47	42
22	47	42
24	48	42
20	49	41
22	49	40
24	47	38
22	49	37
24	49	36
24	50	36
23	49	37
24	50	37
23	49	36
23	52	36

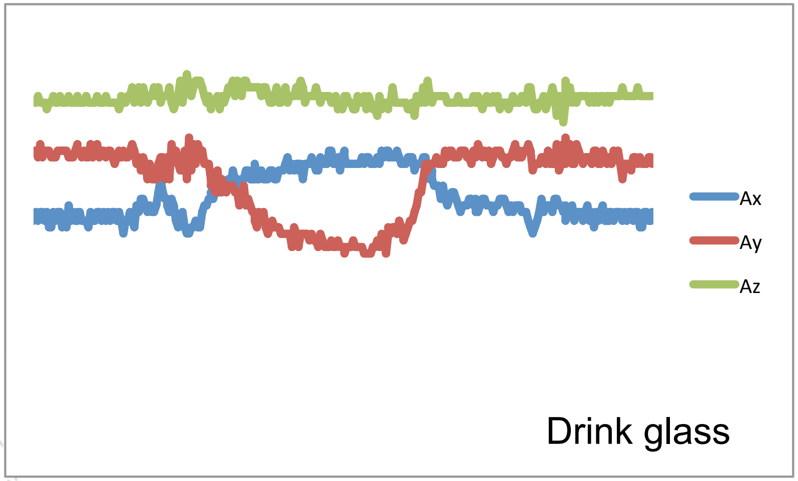
Raw data from accelerometer for the task “brush_teeth”. The columns represents the acceleration measured in X,Y and Z direction.

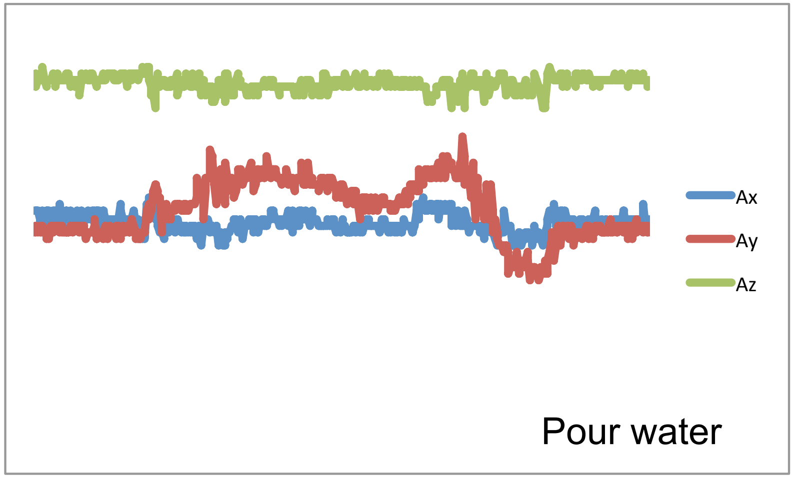
Feature extraction

- A key challenge for any classification ML algorithm is feature extraction -.i.e. unique parameters that distinguishes each class (Activity)









Feature extraction approaches

Group	Methods
Time domain	Mean [1], Std. Deviation [2], Variance [3], MAD [4], Entropy [5]
Frequency domain	Fast Fourier transform [6], Discrete cosine transform [7]
Other	Principal component analysis [8], Linear discriminant analysis [9], Singular value decomposition

- The ML algorithm implemented for this project uses Discrete cosine transformation and Singular value decomposition approaches to classify ADL.

Data wrangling → **Data exploration** → **Build and select model** → **Validate model**

Data wrangling

- Consolidate into a single CSV file.
- Convert raw data into acceleration and apply filter
- Calculate components of SVD: uvd
- Sum vector magnitude (VM) $=\sqrt{(ax)^2+(ay)^2+(az)^2}$

Data exploration

- VM and SVD histograms
- Scale data
- Under-sample and Oversample data
- Split data

Build and select model

- Build model in SVM using training data
- Calculate values for Cost and Gamma
- Cross validate model on training data.

Validate model

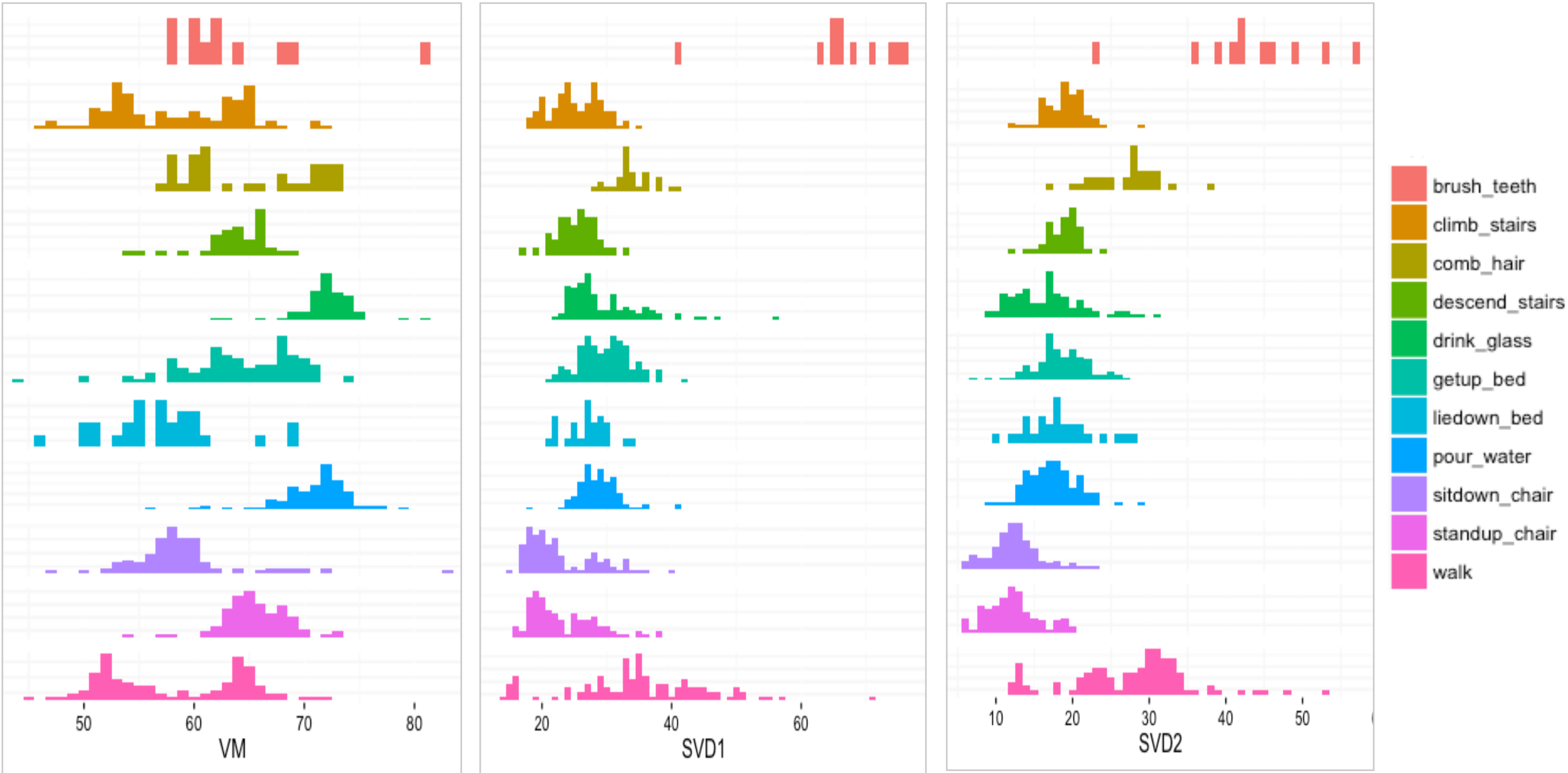
- Use test data to validate model.
- F measure
- Accuracy
- Sensitivity
- Selectivity

CONSOLIDATING INFORMATION

- The original data comprises 479,288 observations of ax, ay, az, distributed among the 11 Activities
- The data is consolidated to 827 observations that contribute to maximum variance in the data.
    - VM refers to sum of vector magnitude
    - SVD 1 and SVD2 refer to d1 and d2 components of Singular value decomposition
    - Act refers to the activities

```
> names(dcs_comb)
[1] "VM"   "SVD1" "SVD2" "Act"
> summary(dcs_comb)
       VM             SVD1             SVD2                        Act
 Min.   :44.16   Min.   :14.35   Min.   : 5.503   walk            :110
 1st Qu.:58.27   1st Qu.:23.47   1st Qu.:13.552   climb_stairs :102
 Median :64.27   Median :27.49   Median :17.621   standup_chair:102
 Mean   :63.69   Mean   :28.40   Mean   :18.553   getup_bed     :101
 3rd Qu.:69.50   3rd Qu.:31.67   3rd Qu.:21.187   drink_glass   :100
 Max.   :83.34   Max.   :76.09   Max.   :57.454   pour_water    :100
                                                  (Other)       :212
> summary(dcs_comb$Act)
  brush_teeth   climb_stairs      comb_hair descend_stairs    drink_glass      getup_bed    liedown_bed     pour_water
           11            102             31             42            100            101             28            100
sitdown_chair  standup_chair           walk
          100            102            110
> str(dcs_comb)
'data.frame':   827 obs. of  4 variables:
 $ VM  : num  67.8 62.5 63.7 61.1 60.2 ...
 $ SVD1: num  40.6 64.5 64.7 62.6 70.8 ...
 $ SVD2: num  23.5 39.2 36.3 57.5 46.2 ...
 $ Act : Factor w/ 11 levels "brush_teeth",..: 1 1 1 1 1 1 1 1 1 1 ...
```

What does the data look like?



Histogram of VM

Histogram of SVD1

Histogram of SVD2

Two models are evaluated

1. Model 1 uses VD as the dependent variable to predict ADL
2. Model 2 uses VD and SVD to predict ADL

DATA SAMPLING

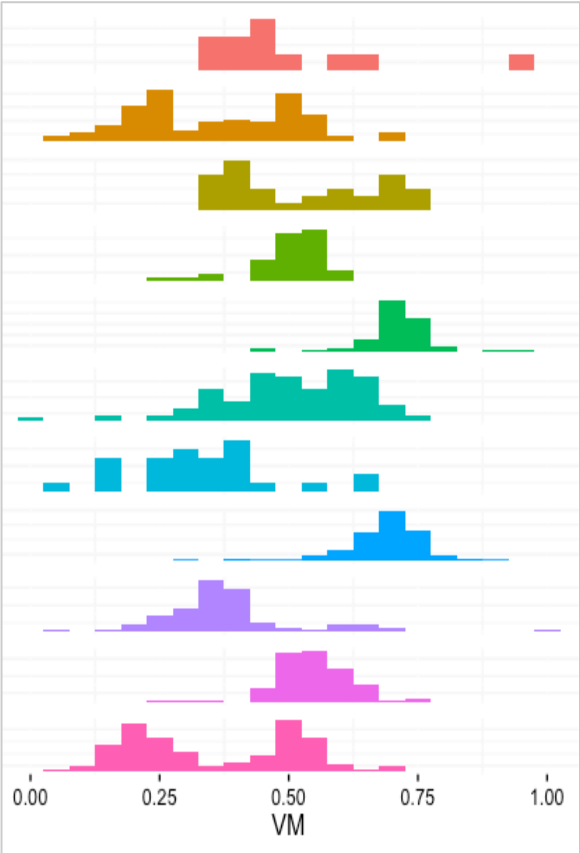
- The original data set is unbalanced with fewer data points for activities such as brush_teeth, comb_hair, and liedown_bed

```
> summary(dcs_comb$Act)
  brush_teeth   climb_stairs       comb_hair descend_stairs     drink_glass       getup_bed     liedown_bed      pour_water
           11            102              31             42             100             101              28             100
sitdown_chair  standup_chair            walk
          100            102             110
```

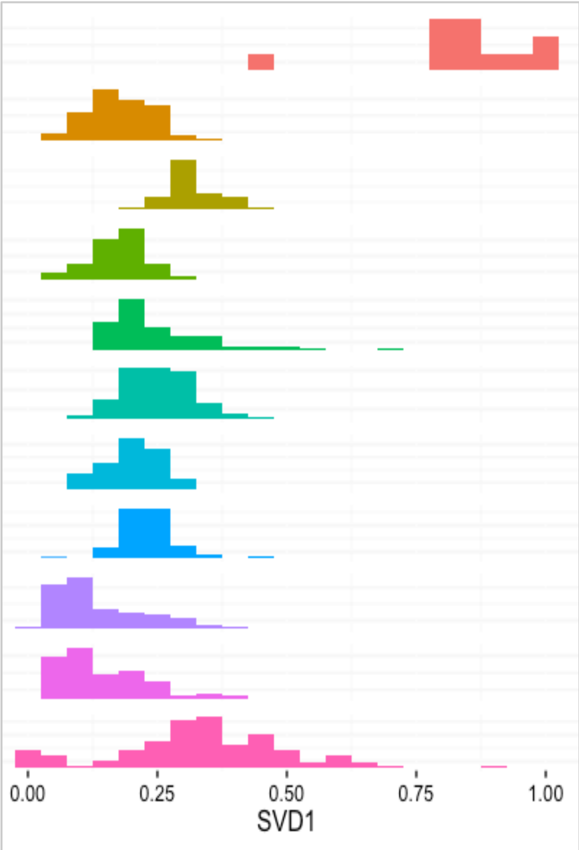
- The SMOTE function is used to handle unbalanced classification problem. It generates **synthetic data sets** that addresses the class unbalance problem.
- Following which the data set is split into training (70%) and test data (30%)

```
> summary(files_smote$Act)
  brush_teeth   climb_stairs       comb_hair descend_stairs     drink_glass       getup_bed     liedown_bed      pour_water
          121            201              65             87             198             187              60             201
sitdown_chair  standup_chair            walk
          207            207             237
> summary(traindata$Act)
  brush_teeth   climb_stairs       comb_hair descend_stairs     drink_glass       getup_bed     liedown_bed      pour_water
           85            141              46             61             139             131              42             141
sitdown_chair  standup_chair            walk
          145            145             166
> summary(testdata$Act)
  brush_teeth   climb_stairs       comb_hair descend_stairs     drink_glass       getup_bed     liedown_bed      pour_water
           36             60              19             26              59              56              18              60
sitdown_chair  standup_chair            walk
           62             62              71
```

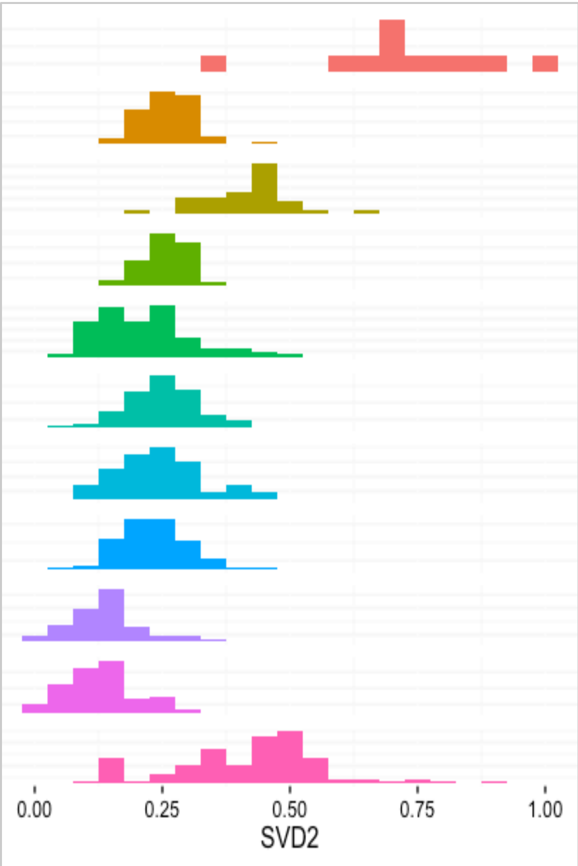
Data after balancing and scaling



Histogram of VM



Histogram of SVD1

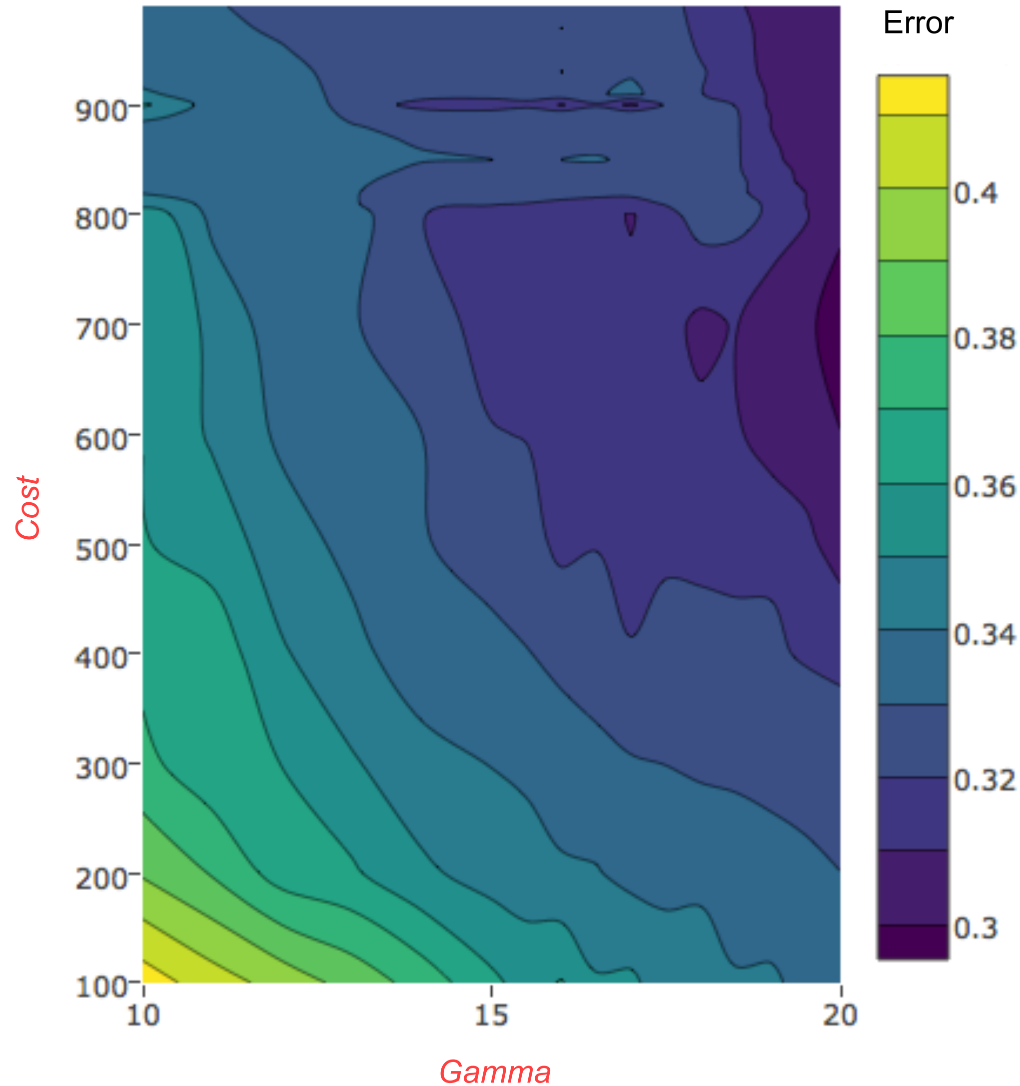


Histogram of SVD2

Scaling avoids attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is it reduces numerical difficulties during the calculation

Tuning of SVM parameters- for Model 1(VD only)

- RBF kernel is chosen as it can handle non-linear relations between class and attributes
- Cost (C) and Gamma (γ) parameters are the two key parameters for the SVM model
- Initial values of cost and gamma for the SVM model are estimated using the "tune function", and running a coarse grid search.
- Following the initial coarse grid search, the value of cost and gamma were fine tuned
- The tune function implemented uses a 10 fold cross validation
- Parameter tuning of 'svm':
  - Sampling method: 10-fold cross validation
  - Best parameters:
    - Gamma:20
    - cost:900
  - Best performance: 0.715 (1 – error)
- The model performance is first evaluated on training data set, followed by test data.

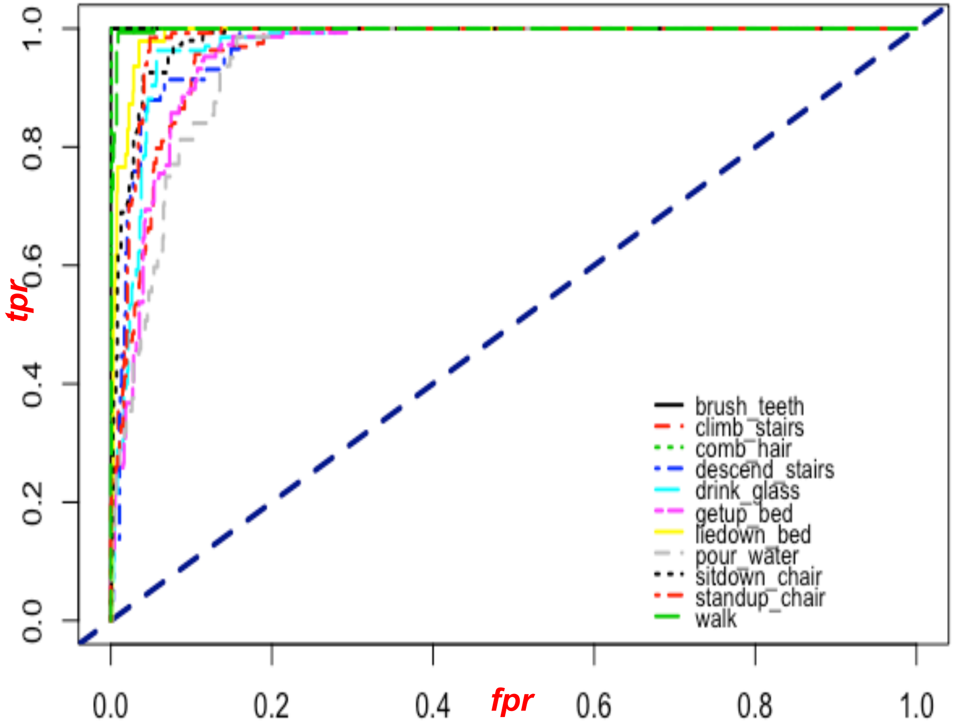


Contour map showing the error rate for different cost and gamma values

Model performance

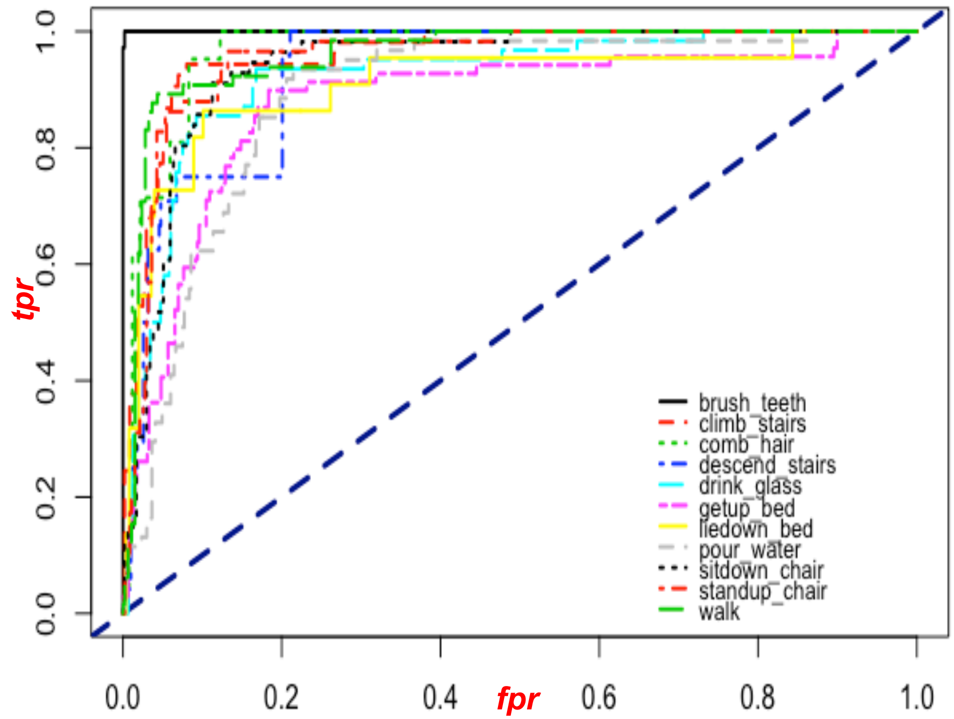
ROC curves and Confusion matrix (CM)

ROC and CM for train data set



TRAINING		CONFUSION MATRIX										
Reference	Prediction	brush_ teeth	climb_ stairs	comb_ hair	Descend _stairs	drink_ glass	getup_ bed	liedown_ bed	pour_ water	sitdown_ chair	Standup _chair	walk
85	brush_teeth	85	0	0	0	0	0	0	0	0	0	0
138	climb_stairs	0	118	0	13	5	9	2	7	0	1	0
47	comb_hair	0	0	42	0	0	0	0	0	0	0	0
60	descend_stairs	0	2	0	35	0	5	0	3	0	0	0
135	drink_glass	0	2	0	1	123	1	0	2	1	1	0
153	getup_bed	0	3	0	3	5	104	2	9	1	2	0
40	liedown_bed	0	2	0	0	1	2	35	0	0	0	0
147	pour_water	0	14	0	7	5	7	2	120	2	1	2
136	sitdown_chair	0	0	0	1	0	1	0	0	129	20	0
144	standup_chair	0	0	0	1	0	0	1	0	12	120	0
161	walk	0	0	4	0	0	2	0	0	0	0	164

ROC and CM for test data set



TEST DATA SET		CONFUSION MATRIX										
Reference	Prediction	brush_ teeth	climb_ stairs	comb_ hair	descend_ stairs	drink_ glass	getup_ bed	liedown_ bed	pour_ water	sitdown_ chair	Standup _chair	walk
36	brush_teeth	34	0	0	0	0	0	0	0	0	0	0
60	climb_stairs	0	42	0	4	2	5	0	6	0	0	1
19	comb_hair	0	0	14	0	2	0	0	0	0	0	1
26	descend_stairs	0	1	0	10	2	2	0	2	0	0	0
59	drink_glass	0	2	1	1	41	3	0	3	2	2	0
56	getup_bed	0	1	0	2	9	33	0	7	1	0	0
18	liedown_bed	0	4	0	0	0	3	15	1	2	0	0
60	pour_water	0	6	0	7	1	7	0	41	1	0	3
62	sitdown_chair	0	1	0	1	0	1	0	0	47	11	0
62	standup_chair	0	2	0	0	1	0	3	0	9	49	0
71	walk	2	1	4	1	1	2	0	0	0	0	66

VD model test summary

- The initial results using VD' to predict ADL were satisfactory, but showed deviation from training data set when implemented on test model.
- The training model performance could be further improved by tuning the value of cost and gamma ,.i.e. high cost and gamma number.
- Prior to implementing model 2, we need to check for collinearity (Strong correlation between two or more predictor variables).
- The vif function is used to check for collinearity among the three variables

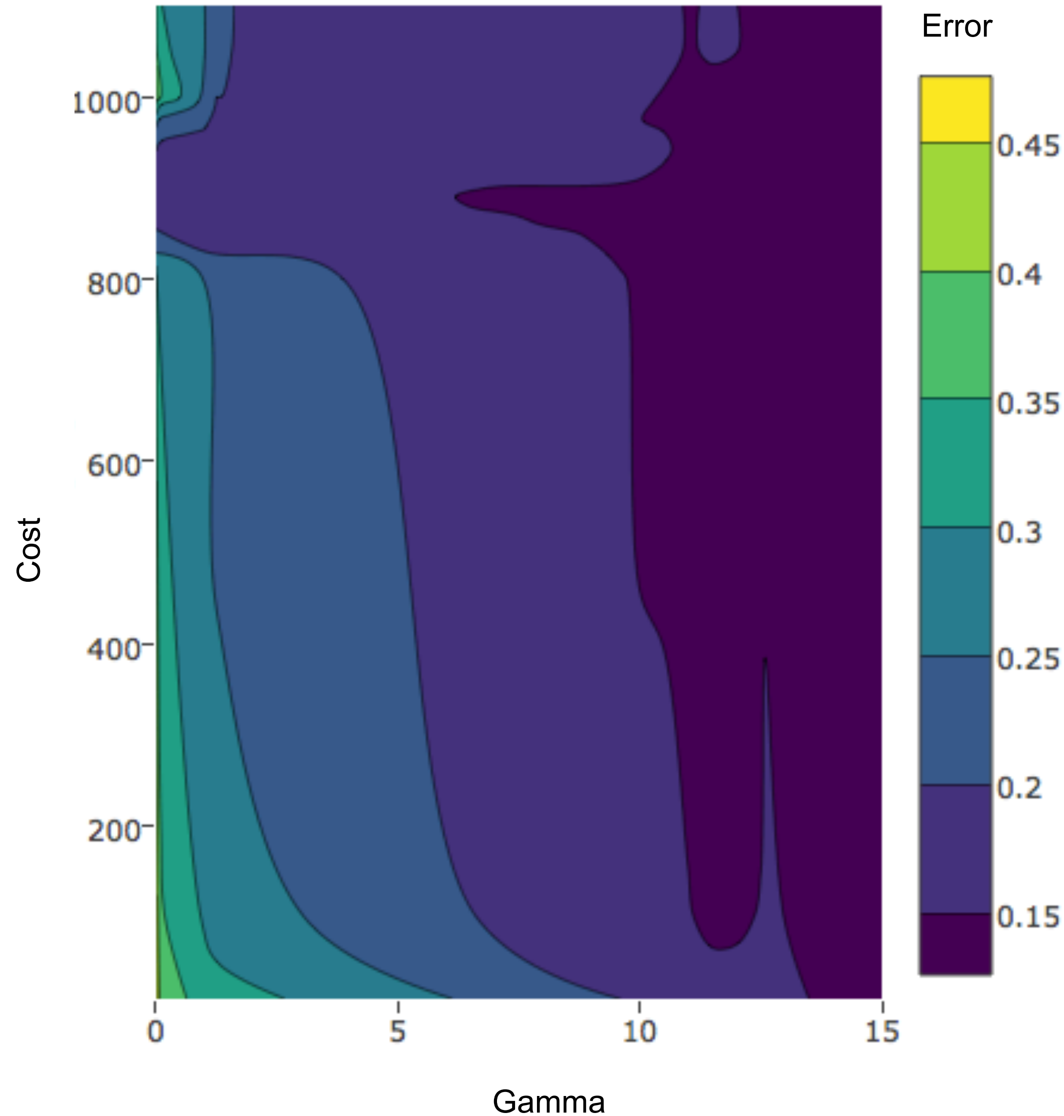
```
> vif(files_dcs[,-4])
  Variables      VIF
1       SVM 1.153859
2      SVD1 2.397817
3      SVD2 2.417696
> vifcor(files_dcs[,-4])
No variable from the 3 input variables has collinearity problem.

The linear correlation coefficients ranges between:
min correlation ( SVD1 ~ SVM ):  0.1195504
max correlation ( SVD2 ~ SVD1 ):  0.7277124

---------- VIFs of the remained variables --------
  Variables      VIF
1       SVM 1.153859
2      SVD1 2.397817
3      SVD2 2.417696
```

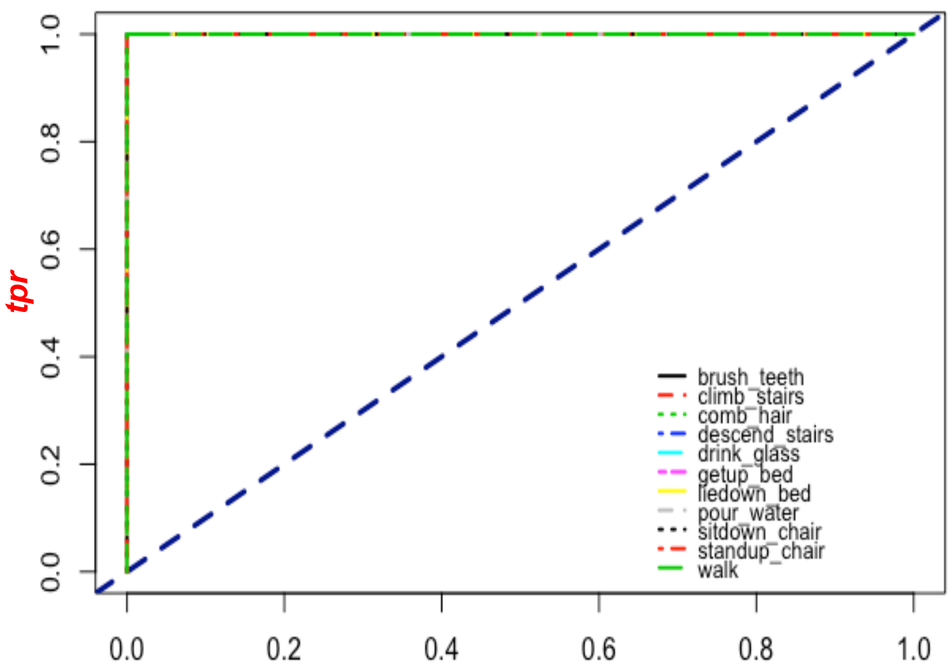
Tuning of SVM parameters: Model 2 (VD and SVM)

- The initial value of cost and gamma for the SVM model are predicted using the “tune function”, and running a coarse grid search.
- Following the initial coarse grid search, the value of cost and gamma were fine tuned
- Each set of cost and gamma values are cross validated (10 fold)
- Parameter tuning of ‘svm’:
  - Sampling method: 10-fold cross validation
  - Best parameters:
    - gamma:18
    - cost:800
  - best performance: 0.863 (1 – error)



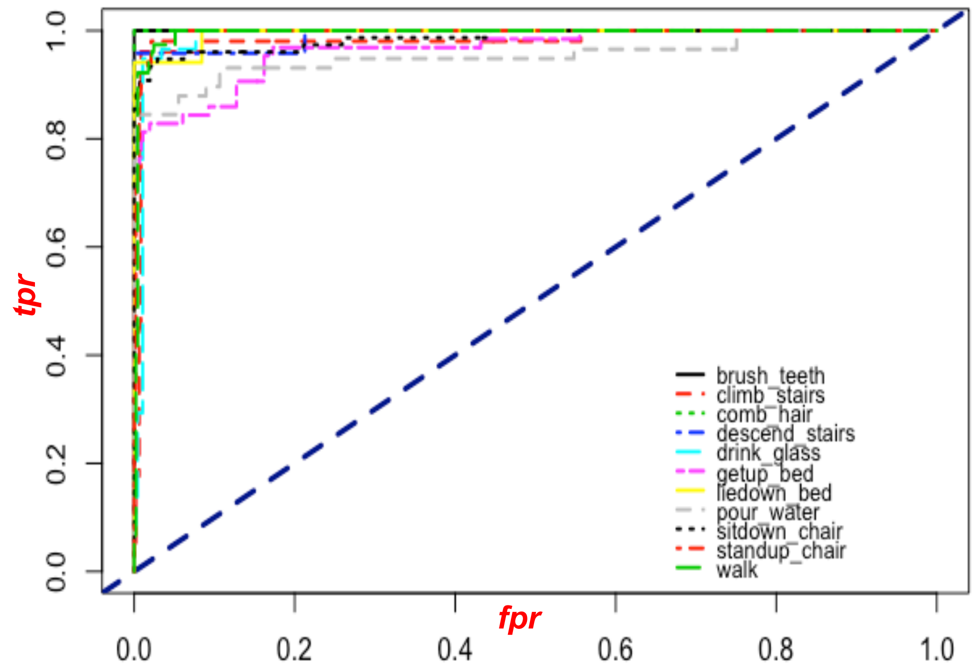
ROC curves and Confusion matrix (CM)

ROC and CM for train data set



TRAINING DATA SET												
Reference	Prediction	brush_ teeth	climb_ stairs	comb_ hair	descend_ stairs	drink_ glass	getup_ bed	liedown_ bed	pour_ water	sitdown_ chair	standup_c hair	walk
85	brush_teeth	85	0	0	0	0	0	0	0	0	0	0
138	climb_stairs	0	152	0	0	0	0	0	0	0	0	0
47	comb_hair	0	0	50	0	0	0	0	0	0	0	0
60	descend_stairs	0	0	0	54	0	0	0	0	0	0	0
135	drink_glass	0	0	0	0	154	0	0	0	0	0	0
153	getup_bed	0	0	0	0	0	134	0	0	0	0	0
40	liedown_bed	0	0	0	0	0	0	36	0	0	0	0
147	pour_water	0	0	0	0	0	0	0	127	0	0	0
136	sitdown_chair	0	0	0	0	0	0	0	0	140	0	0
144	standup_chair	0	0	0	13	0	0	0	0	0	146	0
161	walk	0	0	0	0	0	0	0	0	0	0	153

ROC and CM for test data set



TEST DATA SET												
Reference	Prediction	brush_ teeth	climb_ stairs	comb_ hair	Descend _stairs	drink_ glass	getup_ bed	liedown_ bed	pour_ water	sitdown_ chair	Standup _chair	walk
36	brush_teeth	35	0	0	0	0	0	0	0	0	0	0
58	climb_stairs	0	49	0	0	0	1	0	2	1	1	3
15	comb_hair	0	0	15	0	0	1	0	0	0	0	0
25	descend_stairs	0	1	0	23	0	0	0	0	0	1	0
57	drink_glass	0	0	0	0	56	3	0	7	0	0	0
65	getup_bed	0	0	0	0	1	53	0	3	2	0	1
16	liedown_bed	0	0	0	0	0	0	16	0	1	0	0
62	pour_water	0	0	0	0	0	0	0	46	0	0	0
57	sitdown_chair	0	0	0	0	0	0	0	0	66	0	0
61	standup_chair	0	0	0	1	0	0	0	0	3	47	0
69	walk	1	1	0	0	1	6	1	0	3	2	73

Summary

- The results from the study demonstrate a improvised method for detecting ADL'
- The VD and SVM factors when used in conjunction help to classify activities that would otherwise be misclassified.
- The study needs to extended to other ADL data sets that are publically available.
- While publically available data sets will help tune and validate the model, the efficacy of the model can only be validated using data sets from the industry.
- The application of a machine learning algorithm in this setting is quite vast
  - VR band
  - Health monitoring
  - Activity monitoring
  - Quality of activity, not just quantity
  - Remote point of care devices
  - Fall sensors

References

1. J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korho- nen, “Activity classification using realistic data from wearable sensors,” IEEE Transactions on Information Technology in Biomedicine, vol. 10, no. 1, pp. 119–128, 2006.

2. E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, “Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart monitor,” in *International Symposium on Wearable Computers*, 2007.

3. O. D. Lara, A. J. Perez, M. A. Labrador, and J. D. Posada, “Centinela: A human activity recognition system based on acceleration and vital sign data,” Journal on Pervasive and Mobile Computing, 2011.

4. Y.-P. Chen, J.-Y. Yang, S.-N. Liou, Gwo-Yun=Lee, and J.-S. Wang, “Online classifier construction algorithm for human activity detection using a tri-axial accelerometer,” *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 849–860, 2008.

5. M. Ermes, J. Parkka, and L. Cluitmans, “Advancing from offline to online activity recognition with wearable sensors,” in *Engineering in Medicine and Biology Society. 30th Annual International Conference of the IEEE*, pp. 4451–4454, 2008.

6. L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data,” in *Pervasive*, pp. 1–17, 2004.

7. K. Altun and B. Barshan, “Human activity recognition using iner- tial/ magnetic sensor units,” in Human Behavior Understanding, Lecture Notes in Computer Science, pp. 38–51, Springer Berlin / Heidelberg, 2010.

8. Z. He and L. Jin, “Activity recognition from acceleration data based on discrete consine transform and svm,” in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 5041–5044, 2009.

9. Z. He, Z. Liu, L. Jin, L.-X. Zhen, and J.-C. Huang, “Weightlessness feature; a novel feature for single tri-axial accelerometer based activity recognition,” in 19th International Conference on Pattern Recognition, pp. 1–4, 2008.

10. Y.-P. Chen, J.-Y. Yang, S.-N. Liou, Gwo-Yun=Lee, and J.-S. Wang, “Online classifier construction algorithm for human activity detection using a tri-axial accelerometer,” Applied Mathematics and Computation, vol. 205, no. 2, pp. 849–860, 2008.

11. Chih-Wei Hsu,Chih-Chung Chang, and Chih-Jen Lin, “A Practical Guide to Support Vector Classification”, Department of Computer Science,National Taiwan University, Taipei 106, Taiwan

12. Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition, 30 (7), 1145-1159.

13. Tobias Sing, Oliver Sander, Niko Beerenwinkel, Thomas Lengauer, ROCR: visualizing classifier performance in R.*Bioinformatics* 21(20): 3940-3941 (2005).

14. A Survey on Human Activity Recognition using Wearable Sensors, O Óscar D.Lara and Miguel A. Labrador, Department of Computer Science and Engineering University of South Florida, Tampa, FL 33620

15. David J. Hand and Robert J. Till (2001). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning* **45**(2), p. 171—186.