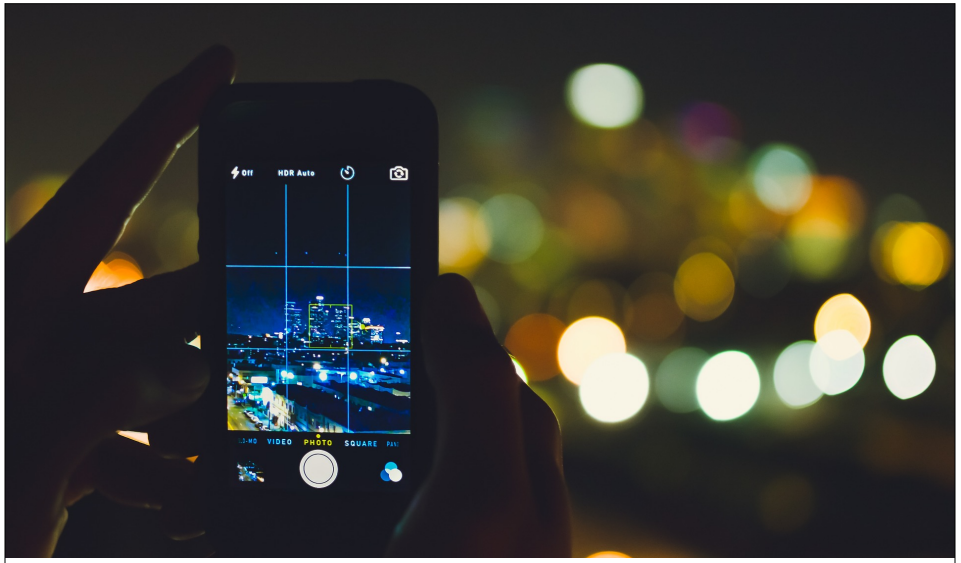


Conception de logiciels Adaptés
Contexte & téléphone intelligent

Sébastien Mosser

INF600G - E20 - Séquence 3- Partie 3

UQÀM | Département d'informatique

Crédit Images: Pixabay & Pexels

ace

Remerciements

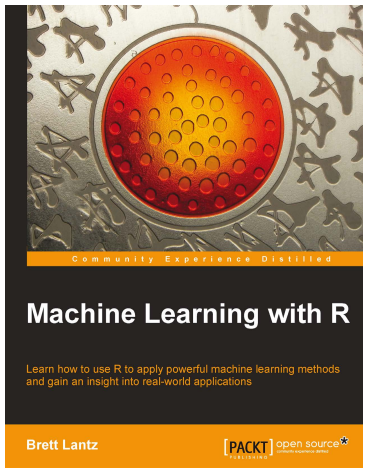
Pr. Arthur Charpentier
Département de Maths

Pr. Marie-Jean Meurs
Département Informatique

Dr. Yonatan Vaizman
UCSD

Ceci n'est TOUJOURS PAS un cours d'IA

Bibliographie



[Lantz, 2013]

Recognizing Detailed Human Context In-the-Wild from Smartphones and Smartwatches

Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet, *senior member, IEEE*

This version was accepted to IEEE Pervasive Computing magazine.
Please cite the official (edited and stylized) paper, which appears in IEEE Pervasive Computing, vol. 16, no. 4, October-December 2017, pp. 62-74.

Abstract—The ability to automatically recognize a person's behavioral context can contribute to health monitoring, aging care and many other domains. Validating context recognition *in-the-wild* is crucial to promote practical applications that work in real-life settings. We collected over 300k minutes of sensor data with context labels from 60 subjects. Unlike previous studies, our subjects used their own personal phone, in any way that was convenient to them, and engaged in their routine in their natural environments. Unscripted behavior and unconstrained phone usage resulted in situations that are harder to recognize. We demonstrate how fusion of multi-model sensors is important for resolving such cases. We present a baseline system, and encourage researchers to use our public dataset to compare methods and improve context recognition in-the-wild.

Index Terms—Context awareness, Mobile sensors, Artificial intelligence, Machine learning, Human activity recognition.

1 Introduction

The ability to automatically recognize a person's context (i.e., where they are, what they are doing, who they are with, etc.) is greatly beneficial in many domains. Health monitoring and lifestyle interventions have traditionally been based on manual, subjective reporting [1], sometimes by end-of-day recalling [2]. These can improve with automatic (frequent, effortless, and objective) detection of behaviors like exercise, eating, sleeping, or mental states like stress. Just-in-time interventions (e.g. for addiction) often prompt the patient at arbitrary times of the day, possibly missing times when the patient needs support the most [3]. Automatically recognizing context will help detect critical times and offer immediate support (e.g. an alcoholic patient may be in high risk of craving or lapse when the context is "at a bar, with friends").

The biomedical research community acknowledges the effects of behavior, lifestyle and environment on health, disease and treatments [4]. Automatic context recognition tools will be essential to incorporate behavioral and exposure aspects into large scale studies and to tailor appropriate treatment for patients. The range of measured exposures should be broad and cover diverse life style and environmental aspects. Commercial tools that offer superficial recognition (e.g. of walking, running, and driving) will not suffice. Personal assistant systems can adjust to context and better serve the user. Aging care programs can use automated logging of older adults' behavior to detect early signs of cognitive impairment, monitor functional independence, and support aging at home [5].

In order for such applications to succeed in large scale, the context recognition component has to be unobtrusive and to work smoothly, without requiring the person to adjust their behavior. It is important that research emulates real-world settings, where such applications will eventually be deployed. In this work we promote context recognition *in-the-wild*, meaning capturing people's authentic behavior in their natural environments, with the use of every-day devices — smartphones and smartwatches. We address the difficulty that in-the-wild conditions add, and show how multi-modal sensors can help.

Related work

It is common for people to have their phone close to them most of the time [6]. This growing trend, and the variety of built-in sensors, make phones popular agents for recognizing human behavior. Smartwatches are a useful sensing addition. While capturing informative signals about hand and arm motion, they remain very natural to wear and don't add any burden to the user.

Previous works have shown the advantage of fusing sensors of different modalities, from smartphones and smartwatches, to improve recognition of basic movement activities [7] and more complex activities, like smoking or drinking coffee [8]. However, most past works have collected data under heavily controlled conditions, with researchers instructing subjects to perform scripted tasks. Fitting models to recognize prescribed activities may result in poor generalization to real life scenarios [9]. To promote real-life working applications, we argue that research has to be done in natural and realistic settings, satisfying four

[Vaizman et al, 2017]

1. Rétrospective L2
2. Détection de contexte
3. Application au téléphone intelligent
4. Lieux de vie intelligent
5. Travail à faire pour L3

Jeux de données : ExtraSensory

Name	Date Modified	Size	Kind
0A986513-7828-4D53-AA1F-E02D6DF9561B.features_labels.csv	May 15, 2017 at 6:17 PM	8.6 MB	CSV Document
0BFC35E2-4817-4865-BFA7-764742302A2D.features_labels.csv	May 15, 2017 at 6:15 PM	6.7 MB	CSV Document
0E6184E1-90C0-48EE-B25A-F1ECB7B9714E.features_labels.csv	May 15, 2017 at 6:16 PM	15.6 MB	CSV Document
00EABED2-271D-49D8-B599-1D4A09240601.features_labels.csv	May 15, 2017 at 6:14 PM	4.5 MB	CSV Document
1DBB0F6F-1F81-4A50-9DF4-CD62ACFA4842.features_labels.csv	May 15, 2017 at 6:14 PM	15.9 MB	CSV Document
2C32C23E-E30C-498A-8DD2-0EFB9150A02E.features_labels.csv	May 15, 2017 at 6:13 PM	18.5 MB	CSV Document
4E98F91F-4654-42EF-B908-A3389443F2E7.features_labels.csv	May 15, 2017 at 6:15 PM	6.7 MB	CSV Document
4FC32141-E888-4BFF-8804-12559A491D8C.features_labels.csv	May 15, 2017 at 6:13 PM	10.4 MB	CSV Document
5EF64122-B513-46AE-BCF1-E62AAC285D2C.features_labels.csv	May 15, 2017 at 6:14 PM	8.1 MB	CSV Document
7CE37510-56D0-4120-A1CF-0E23351428D2.features_labels.csv	May 15, 2017 at 6:15 PM	20.6 MB	CSV Document
7D9BB102-A612-4E2A-8E22-3159752F55D8.features_labels.csv	May 15, 2017 at 6:16 PM	3.1 MB	CSV Document
9DC38D04-E82E-4F29-AB52-B476535226F2.features_labels.csv	May 15, 2017 at 6:16 PM	20.4 MB	CSV Document
11B5EC4D-4133-4289-B475-4E737182A406.features_labels.csv	May 15, 2017 at 6:13 PM	19.2 MB	CSV Document
24E40C4C-A349-4F9F-93AB-01D00FB994AF.features_labels.csv	May 15, 2017 at 6:15 PM	10.2 MB	CSV Document
27E04243-B138-4F40-A164-F40B60165CF3.features_labels.csv	May 15, 2017 at 6:16 PM	10.2 MB	CSV Document
33AB5C34-CFE4-4732-9E73-0A7AC861B27A.features_labels.csv	May 15, 2017 at 6:14 PM	13.3 MB	CSV Document
40E170A7-607B-4578-AF04-F021C3B0384A.features_labels.csv	May 15, 2017 at 6:17 PM	14.9 MB	CSV Document
59EEFAE0-DEB0-4FFF-9250-54D2A03D0CF2.features_labels.csv	May 15, 2017 at 6:17 PM	15.9 MB	CSV Document
74B86067-5D4B-43CF-82CF-341B76BEA0F4.features_labels.csv	May 15, 2017 at 6:14 PM	15.1 MB	CSV Document
78A91A4E-4A51-4065-BDA7-94755F0BB3BB.features_labels.csv	May 15, 2017 at 6:16 PM	25.9 MB	CSV Document
83CF687B-7CEC-434B-9FE8-00C3D57998E6.features_labels.csv	May 15, 2017 at 6:16 PM	20.5 MB	CSV Document
86A4F379-B305-473D-9D83-FC7D800180EF.features_labels.csv	May 15, 2017 at 6:13 PM	20.4 MB	CSV Document
96A358A0-FFF2-4239-B93E-C7425B901B47.features_labels.csv	May 15, 2017 at 6:16 PM	12 MB	CSV Document
098A72A5-E3E5-4F54-A152-BBDA0DF7B694.features_labels.csv	May 15, 2017 at 6:13 PM	13.7 MB	CSV Document
99B204C0-DD5C-4BB7-83E8-A372818BD769.features_labels.csv	May 15, 2017 at 6:15 PM	12.1 MB	CSV Document
481F4DD2-7689-43B9-A2AA-C8772227162B.features_labels.csv	May 15, 2017 at 6:17 PM	13.8 MB	CSV Document
797D145F-3858-4A7F-A7C2-A4EB721E133C.features_labels.csv	May 15, 2017 at 6:17 PM	7.5 MB	CSV Document
1155FF54-63D3-4AB2-9863-8385D0BD0A13.features_labels.csv	May 15, 2017 at 6:17 PM	5.6 MB	CSV Document
1538C99F-BA1E-4EFB-A949-6C7C47701B20.features_labels.csv	May 15, 2017 at 6:15 PM	12.7 MB	CSV Document

Macintosh HD › Users › mosser › work › teaching › INF600G › behavioral-context-recognition › _datasets › ExtraSensory

http://extrasensory.ucsd.edu/

Exemple de données collectées

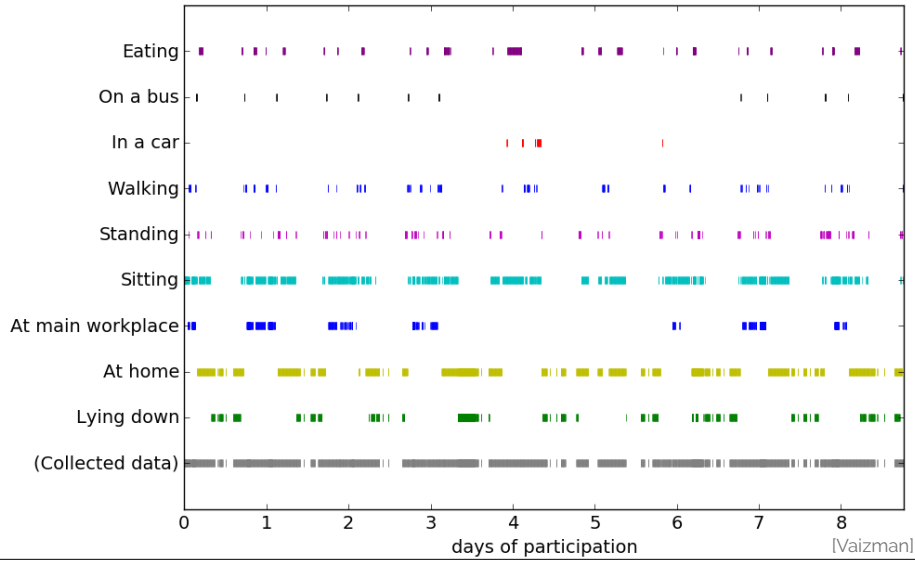
0A986513-7828-4D53-AA1F-E02D6DF9561B.features_labels

Sheet 1

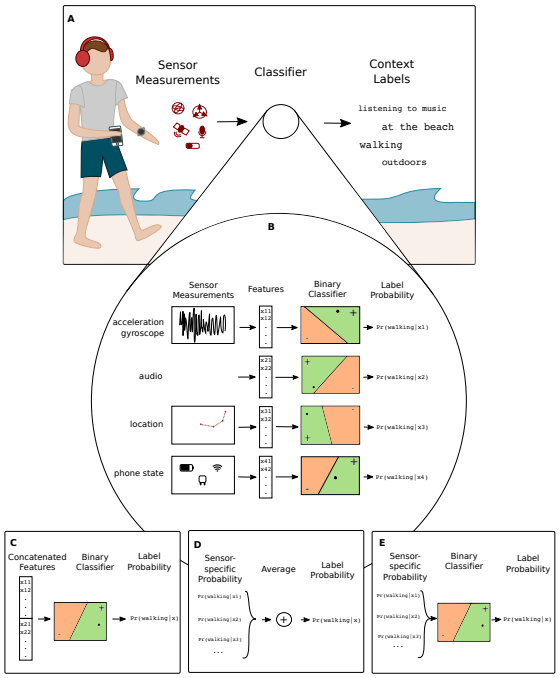
	timestamp	raw_acc:magnitude_stats:mean	raw_acc:magnitude_stats:std	raw_acc:magnitude_stats:moment3	raw_acc:magnitude_stats:moment4	raw
2	1449601597	1.000371	0.007671	-0.016173	0.027860	
3	1449601657	1.000243	0.003782	-0.002713	0.007046	
4	1449601717	1.000811	0.002082	-0.001922	0.003575	
5	1449601777	1.001245	0.004715	-0.002895	0.008881	
6	1449601855	1.001354	0.065186	-0.096520	0.165298	
7	1449601918	1.002472	0.067833	0.164222	0.289580	
8	1449601975	1.001820	0.002009	-0.001022	0.003073	
9	1449602035	1.001889	0.001781	0.000630	0.003392	
10	1449602095	1.004284	0.105347	0.163315	0.295795	
11	1449602155	1.001456	0.003742	-0.004386	0.013095	
12	1449602215	1.001763	0.004617	0.011668	0.020438	

60 participants

Durée de la collecte

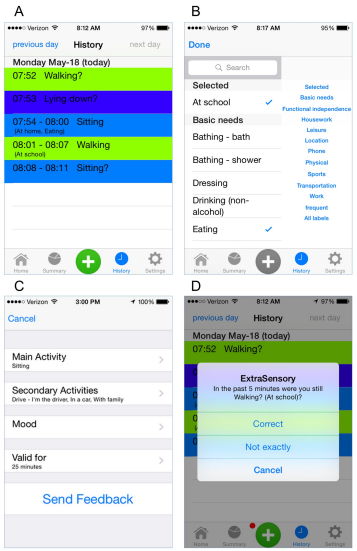


[Vaizman]



[Vaizman et al, 2017]

Application mobile de collecte

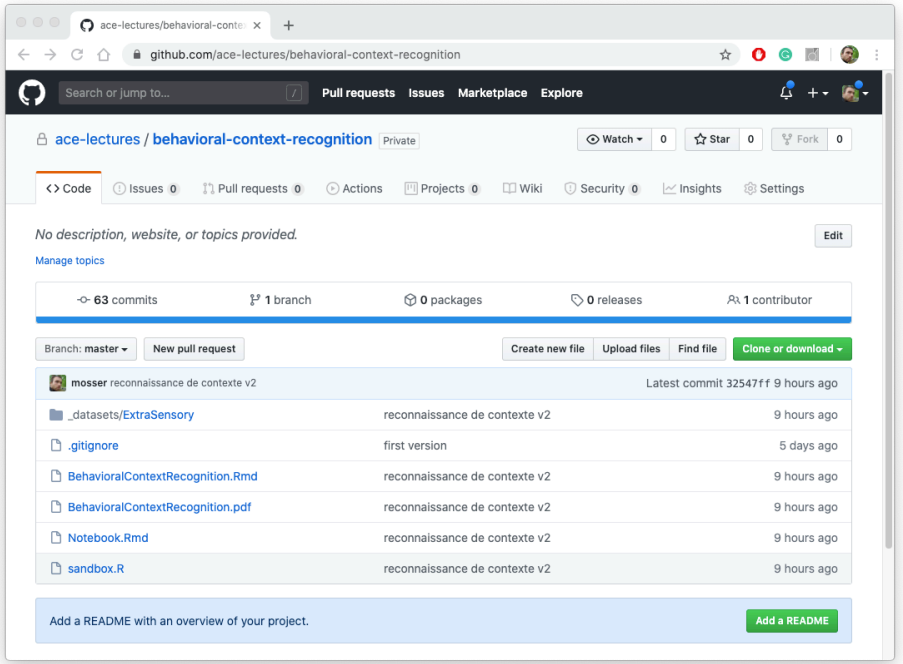


Surveillance du comportement

Annotation par l'utilisateur

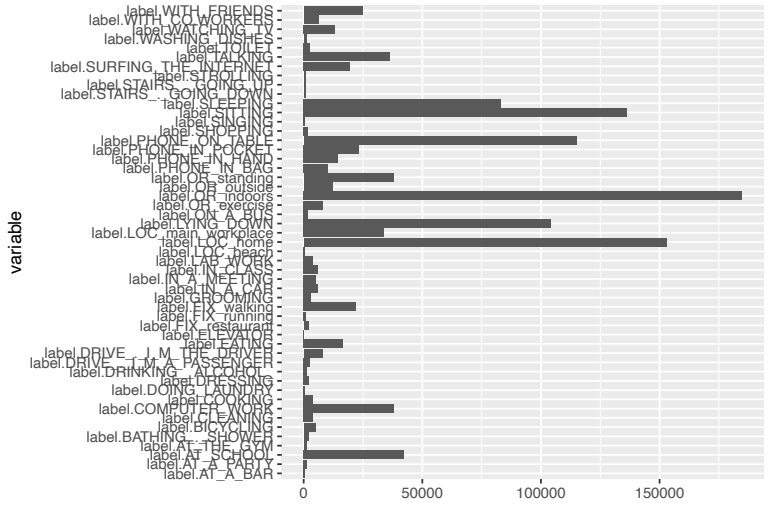
Validation des prédictions

[Vaizman et al, 2017]



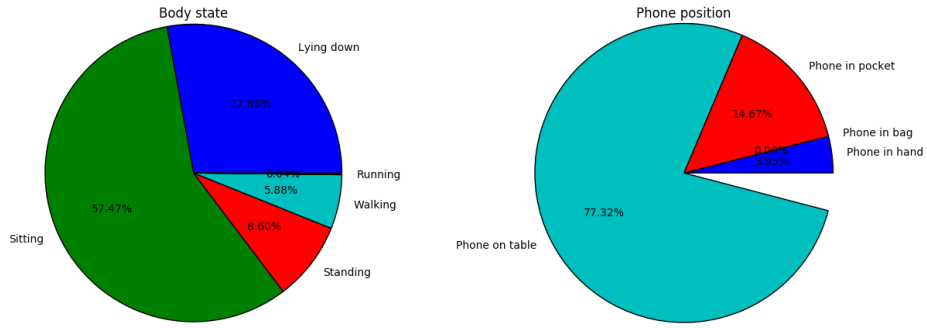
https://github.com/ace-lectures/behavioral-context-recognition

Analyse quantitative du jeux de données

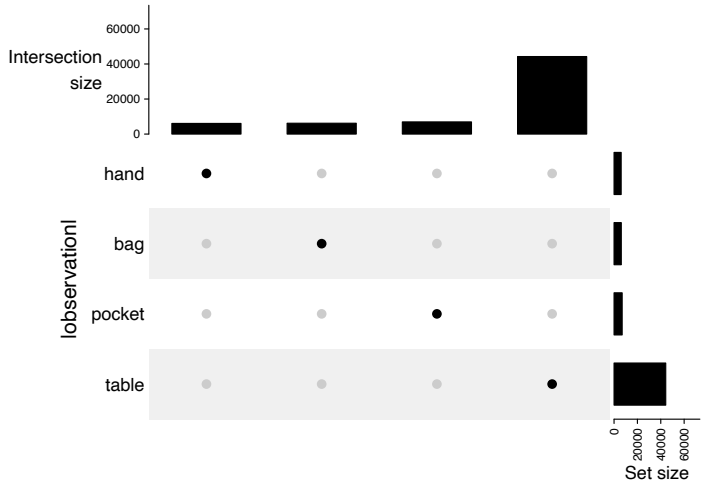


377,346 observations. Espace des données: 278 dimensions

Représentation des observations



Exclusivité des observations



Fabrication d'un jeux de données initial

```
phone_observations <- dataset %>% select(starts_with("raw_acc"),
                                         starts_with("proc_gyro"),
                                         starts_with("raw_magnet"),
                                         starts_with("label:PHONE_"))
phone_observations <- phone_observations[complete.cases(phone_observations),]
```

- Sur les 377,348 observations et 278 dimensions, on garde :
  - Les données de l'accéléromètre, du gyroscope et du magnétomètre
  - Les étiquettes de reconnaissance du téléphone
  - Uniquement les observations complètes pour ces variables là
- Cela nous donne 59,982 observations

Normalisation des données [1/2]

```
##  raw_acc:3d:mean_x   raw_acc:3d:mean_y   raw_acc:3d:mean_z
##  Min.   :-1.134850   Min.   :-1.022699   Min.   :-1.08164
##  1st Qu.:-0.020165   1st Qu.:-0.028909   1st Qu.:-0.99016
##  Median : 0.012589   Median : 0.004741   Median :-0.11415
##  Mean   : 0.005651   Mean   : 0.028688   Mean   :-0.05196
##  3rd Qu.: 0.064781   3rd Qu.: 0.082794   3rd Qu.: 0.96098
##  Max.   : 1.289129   Max.   : 1.290554   Max.   : 1.05319
```

- Les données ne sont pas normalisée
  - et donc difficilement comparables
- On va ramener les données dans un intervalle [0,1]

```
norm_variable <- function (v) { return ((v - min(v)) / (max(v) - min(v))) }
```

Normalisation des données [2/2]

```
##  raw_acc:3d:mean_x   raw_acc:3d:mean_y   raw_acc:3d:mean_z
##  Min.   :-1.134850   Min.   :-1.022699   Min.   :-1.08164
##  1st Qu.:-0.020165   1st Qu.:-0.028909   1st Qu.:-0.99016
##  Median : 0.012589   Median : 0.004741   Median :-0.11415
##  Mean   : 0.005651   Mean   : 0.028688   Mean   :-0.05196
##  3rd Qu.: 0.064781   3rd Qu.: 0.082794   3rd Qu.: 0.96098
##  Max.   : 1.289129   Max.   : 1.290554   Max.   : 1.05319
```

```
normalize <- function(dataset) {
  ds_n <- as.data.frame(lapply(dataset[1:ncol(dataset)-1],
                               norm_variable))
  ds_n$status <- dataset$STATUS
  return(ds_n)
}
```

```
##  raw_acc.3d.mean_x raw_acc.3d.mean_y raw_acc.3d.mean_z
##  Min.   :0.0000    Min.   :0.0000    Min.   :0.00000
##  1st Qu.:0.4599    1st Qu.:0.4296    1st Qu.:0.04285
##  Median :0.4734    Median :0.4442    Median :0.45319
##  Mean   :0.4705    Mean   :0.4545    Mean   :0.48233
##  3rd Qu.:0.4949    3rd Qu.:0.4779    3rd Qu.:0.95680
##  Max.   :1.0000    Max.   :1.0000    Max.   :1.00000
```

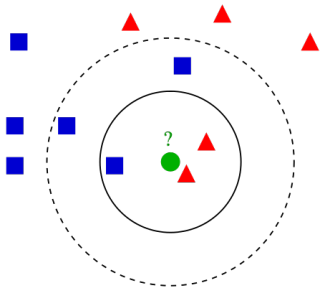
Entrainement & Test

```
prepare_datasets <- function(complete, random.seed=42) {
  set.seed(random.seed) # Pour rendre les experiences reproductibles
  training <- complete %>% sample_n(size = 0.8*nrow(complete))
  test     <- setdiff(complete, training)
  list(train = training, test = test)
}
```

- On va garder 20% des données pour mesurer les performances
- Et entrainer les modèles avec 80%
- *On utilise une "graine" pour rendre les experiences reproductibles*

Un premier prédicteur : kNN

- On applique la méthode des *k-nearest neighbours*:
- Pour une nouvelle observation :
  - On regarde les classes de ses k-plus proches voisins
  - On vote pour la classe majoritaire
- Simple à mettre en place :
  - Distance euclidienne

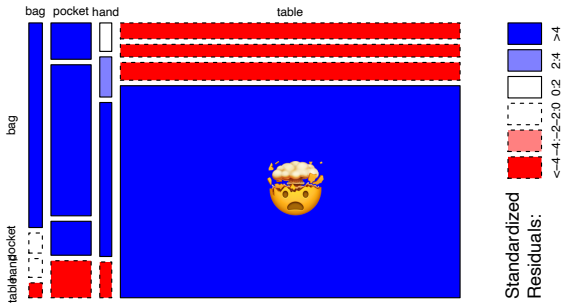


https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Une version de kNN en R

```
run_knn <- function(training, test, k.value=42) {
  # Preparing the datasets for training
  training_data <- training %>% select(-status)
  training_classes <- training %>% select(status)
  # Same, for testing
  test_data <- test %>% select(-status)
  test_classes <- test %>% select(status)
  # Running the classifier
  predictions <- knn(train = training_data,
                     test  = test_data,
                     cl    = training_classes[,1],
                     k     = k.value)
  # Building the confusion matrix
  cmat <- confusionMatrix(reference = test_classes[,1],
                          data = predictions)
  return(cmat)
}
```

Visualisation de la matrice de confusion



	Balanced.Accuracy	Precision	Recall
Class: bag	0.63	0.79	0.28
Class: pocket	0.75	0.58	0.55
Class: hand	0.59	0.60	0.19
Class: table	0.73	0.82	0.97

Un prédicteur encore plus simple : Constant

```
constant_table <- function(training, test) {
  test_classes <- test %>% select(status)
  # On répond toujours table ... facile !
  predictions <- as.factor(rep('table',nrow(test)))
  cmat <- confusionMatrix(reference = test_classes[,1],
                          data = predictions)
  return(cmat)
}
cmat_constant <- with(ds <- prepare_datasets(normalize(phone_observations)),
                      constant_table(ds$train, ds$test))
```

Ce prédicteur à une accuracy de … 70.80% !!!!

On doit nettoyer les données !!

```
summary(phone_observations$STATUS)
```

```
##     bag pocket    hand   table
##    5668    6533    5465   42216
```

- Les observations “téléphone sur la table” sont déséquilibrées
- On va ré-équilibrer le jeu de données :
  - On garde ‘bag’, ‘pocket’ et hand’, environ 6,000 obs/classes
  - On échantillonne (*random*) 6,000 observations pour ‘table’

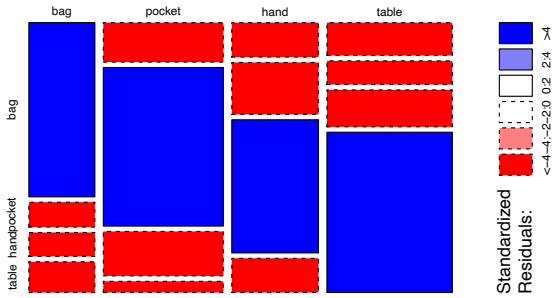
Équilibrage, et ré-apprentissage

```
balance <- function(dataset, random.seed=42) {
  set.seed(random.seed)
  res <- dataset[dataset$STATUS == 'bag',]
  res <- rbind(res, dataset[dataset$STATUS == 'pocket',])
  res <- rbind(res, dataset[dataset$STATUS == 'hand',])
  res <- rbind(res, sample_n(dataset[dataset$STATUS == 'table',], 6000))
  return(res)
}
```

On parle de “pipeline” de traitement de données

```
cmat_balanced <-
  with(ds <- prepare_datasets(normalize(balance(phone_observations))),
                      run_knn(ds$train, ds$test))
```

Résultats sur le jeu équilibré



ACC = 61,75%

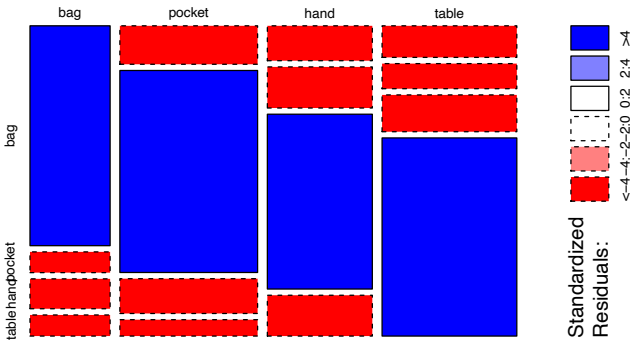
	Balanced.Accuracy	Precision	Recall	F1
Class: bag	0.71	0.69	0.49	0.57
Class: pocket	0.76	0.63	0.68	0.65
Class: hand	0.68	0.52	0.50	0.51
Class: table	0.80	0.63	0.76	0.69

Problème de normalisation

- Notre normalisation "écrase" les valeurs extrêmes
- Ce n'est pas très adapté à des valeurs de capteurs
  - Changement brusque de cap par exemple
- On va utiliser une autre méthode de normalisation : le Z-score
  - Inconvénient : pas dans [0,1] (pas de min ni de max prédéfini)

```
z_normalize <- function(dataset) {
  ds_n <- as.data.frame(scale(dataset[1:ncol(dataset)-1]))
  ds_n$status <- dataset$STATUS
  return(ds_n)
}
```

Matrice de confusion avec Z-score

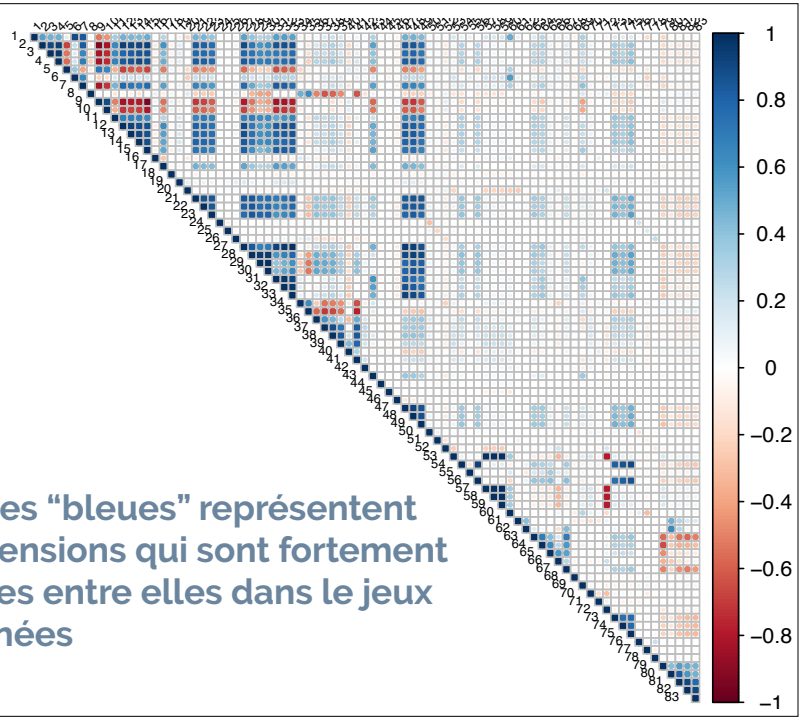


	Balanced.Accuracy	Precision	Recall	F1
Class: bag	0.76	0.75	0.57	0.65
Class: pocket	0.81	0.69	0.75	0.72
Class: hand	0.74	0.60	0.60	0.60
Class: table	0.82	0.68	0.76	0.72

Et si on regardait les dimensions ?

- On travaille avec 83 dimensions sur les observations
- L'algorithme kNN repose sur une distance euclidienne
  - Il est sensible à la présence de "trop" de dimensions
- Peut-on se débarrasser de dimensions corrélées ?

```
print_corplot <- function(dataset) {
  corr_obs <- dataset %>% select(-STATUS)
  colnames(corr_obs) <- 1:ncol(corr_obs)
  corr_matrix <- cor(corr_obs)
  corrplot(corr_matrix, type = "upper",
           tl.col = "black", tl.srt = 45, tl.cex=0.5)
}
```

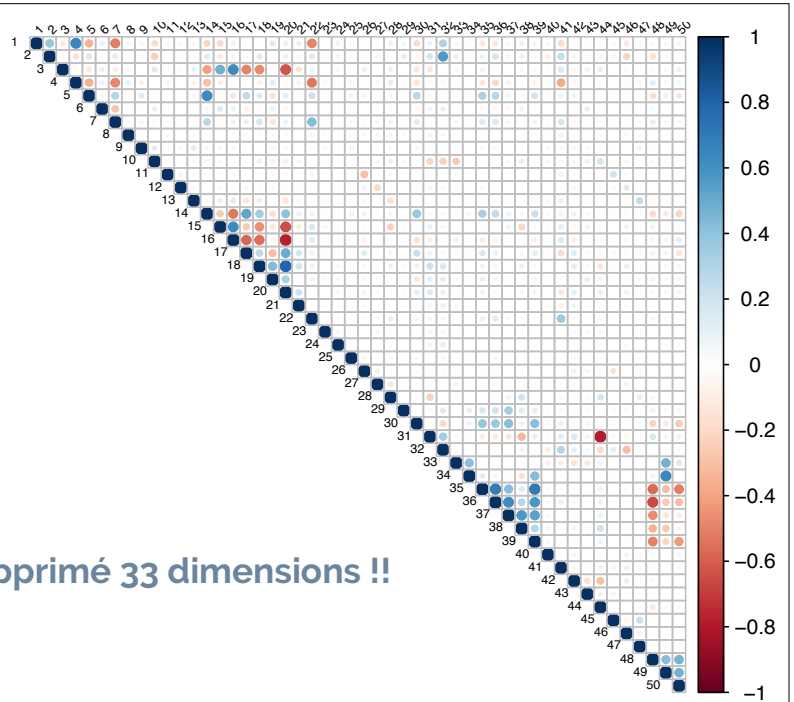


Les cases "bleues" représentent les dimensions qui sont fortement corrélées entre elles dans le jeux de données

Épuration du jeu de données

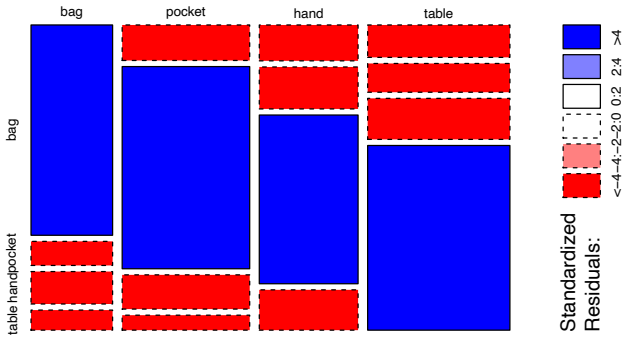
- On va garder uniquement les dimensions “pas trop” corrélées
  - On fixe un seuil arbitraire à 80%
- Cela va grandement accélérer le temps d'entraînement
  - On part d'un espace initial à 83 dimensions …

```
slice_relevant <- function(dataset, threshold = 0.8) {
  corr_obs <- dataset %>% select(-STATUS)
  corr_matrix <- cor(corr_obs)
  high <- findCorrelation(corr_matrix, cutoff = threshold)
  res <- dataset[,-c(high)]
  res$STATUS <- dataset$STATUS
  return(res)
}
print_corplot(slice_relevant(phone_observations))
```



On a supprimé 33 dimensions !!

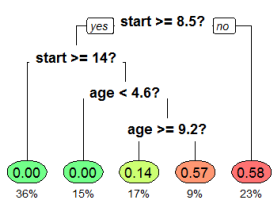
Résultat sur le jeu épuré

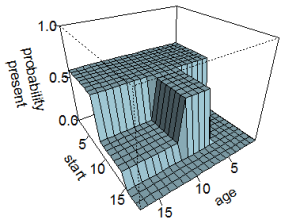


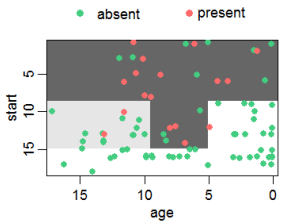
	Balanced.Accuracy	Precision	Recall	F1
Class: bag	0.76	0.73	0.57	0.64
Class: pocket	0.80	0.70	0.72	0.71
Class: hand	0.72	0.59	0.56	0.58
Class: table	0.81	0.64	0.78	0.70

Changement de paradigme : Decision Trees

- Plutôt qu'un seul choix, on va passer par une succession de décisions qui vont nous amener vers la classe de l'élément



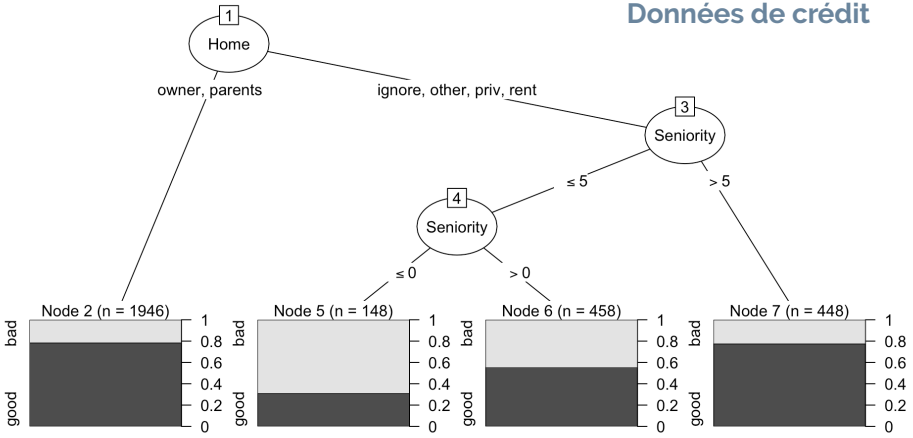




https://en.wikipedia.org/wiki/Decision_tree_learning

Exemple : l'algorithme C5.0

Données de crédit

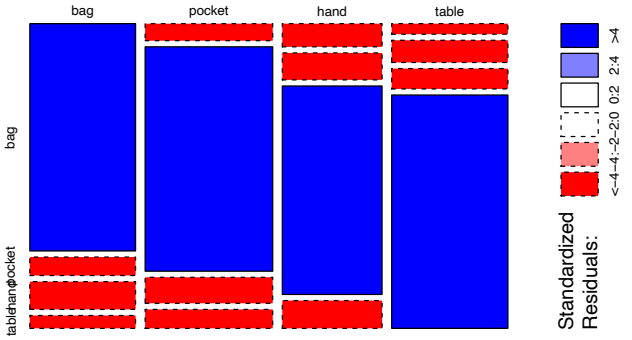


https://cran.r-project.org/web/packages/C50/vignettes/C5.0.html

Entraînement d'un modèle C5.0

```
run_dt <- function(training, test, nb.trials=1) {
  training_data <- training %>% select(-status)
  training_classes <- training %>% select(status)
  test_data <- test %>% select(-status)
  test_classes <- test %>% select(status)
  model <- C5.0(training_data,
                training_classes[,1],
                trials = nb.trials)
  predictions <- predict(model, test_data, type = "class")
  cmat <- confusionMatrix(reference = test_classes[,1],
                          data = predictions)
  return(list(cmat=cmat, tree=model))
```

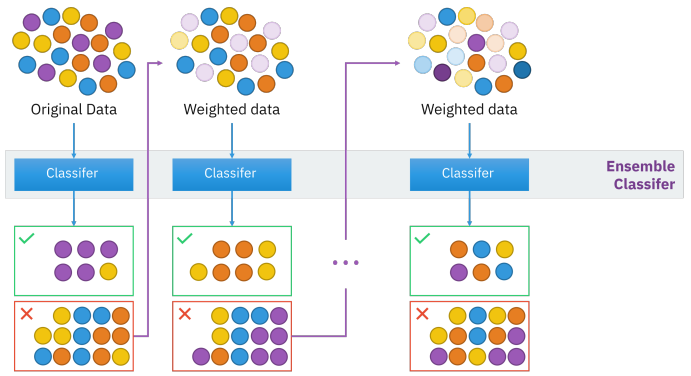
Résultat avec un arbre de décision



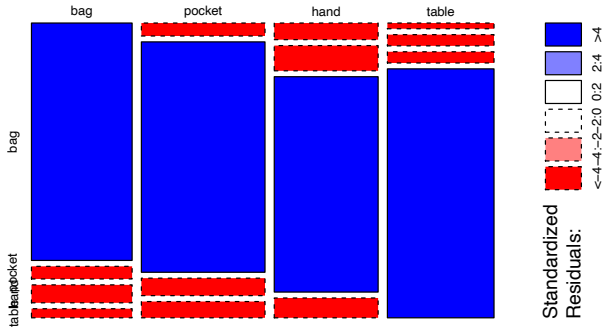
	Balanced.Accuracy	Precision	Recall	F1
Class: bag	0.87	0.79	0.81	0.80
Class: pocket	0.86	0.78	0.80	0.79
Class: hand	0.81	0.73	0.71	0.72
Class: table	0.87	0.81	0.81	0.81

Méta-algorithme: Adaptive Boosting

- Plutôt q'un seul arbre, on va en fabriquer plusieurs
- Les arbres vont se "*spécialiser*" pour trouver les cas "*spéciaux*"

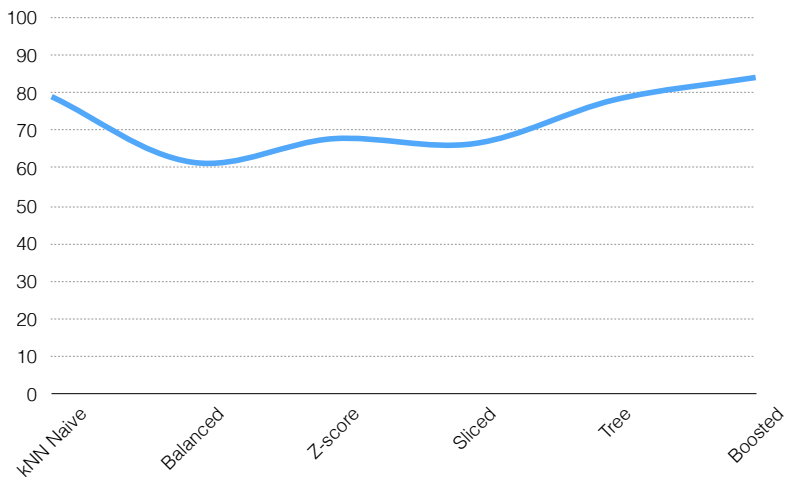


Résultat avec une accélération de 10

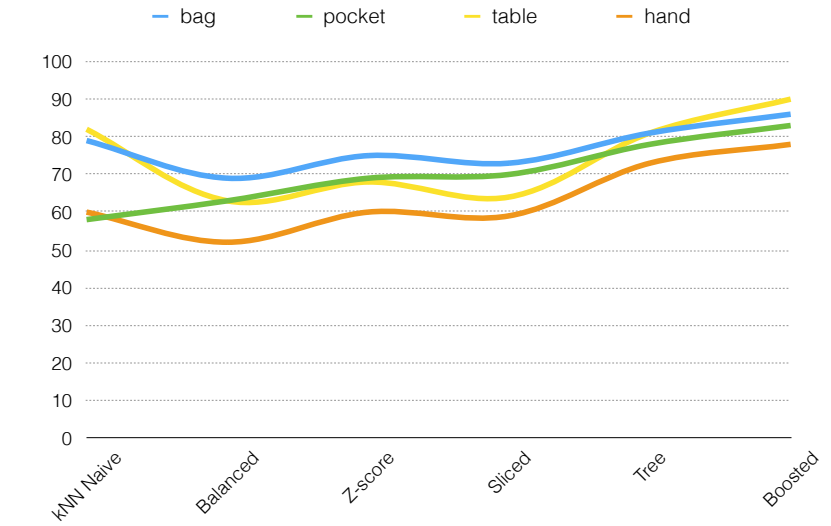


	Balanced.Accuracy	Precision	Recall	F1
Class: bag	0.91	0.86	0.86	0.86
Class: pocket	0.89	0.83	0.85	0.84
Class: hand	0.87	0.78	0.81	0.79
Class: table	0.90	0.90	0.84	0.87

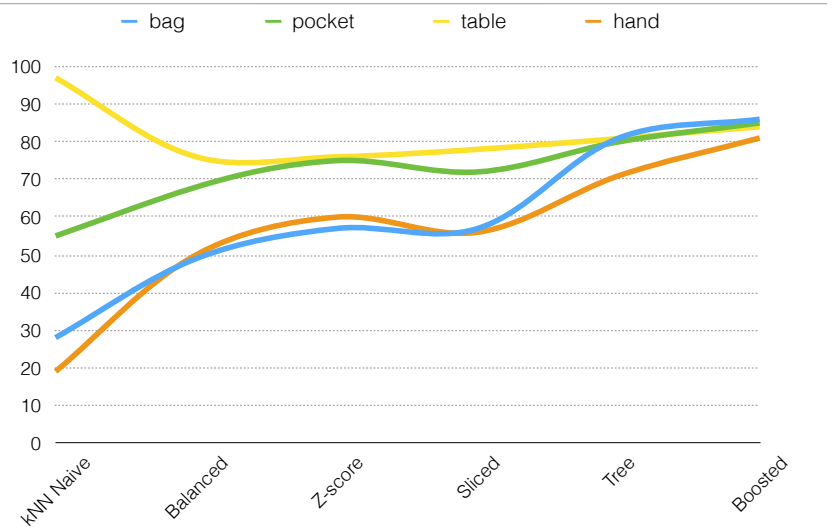
Évolution de l'Accuracy générale



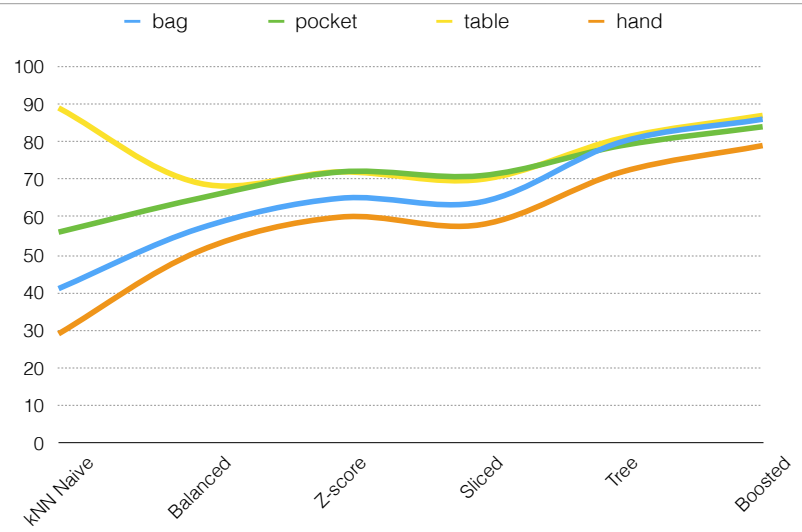
Évolution de la Précision



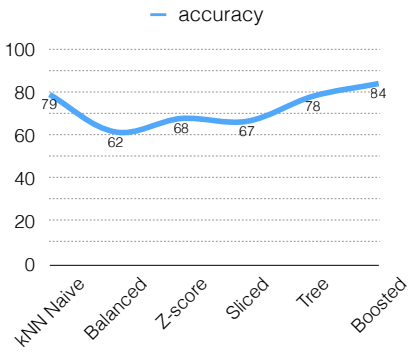
Évolution du Rappel

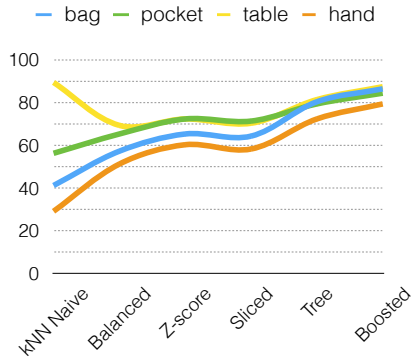


Évolution de la F-mesure



Accuracy versus F-mesure





Faire un prédicteur, c'est simple.
Bien le faire, c'est compliqué …

Publicité (e.g., INF6200)

- En tant que développeur, je fais un commit
- Des modifications concurrentes existent

ace

Quel algorithme de fusion utiliser pour avoir le moins de chances possible de devoir résoudre un "merge conflict" ?