

# Design of a diagnosis and follow-up platform for patients with chronic headaches

Kiani Lannoye & Gilles Vandewiele

Faculty of Engineering and Architecture

Intro

Current process UH Ghent

Platform requirements

Mobile application

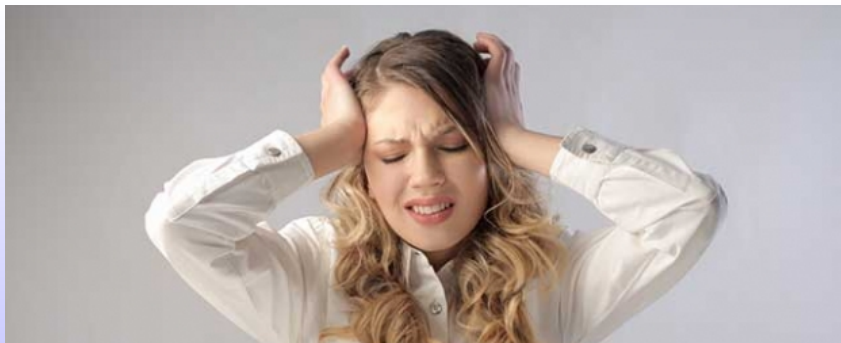
Backend and data exposure

Machine learning

Doctor dashboard

Conclusion & future work

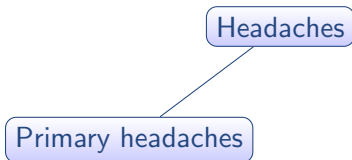
# Headaches



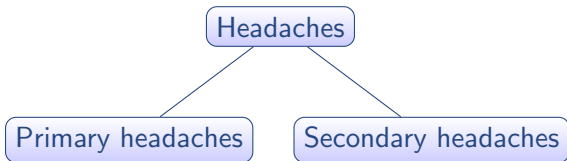
# Headaches

Headaches

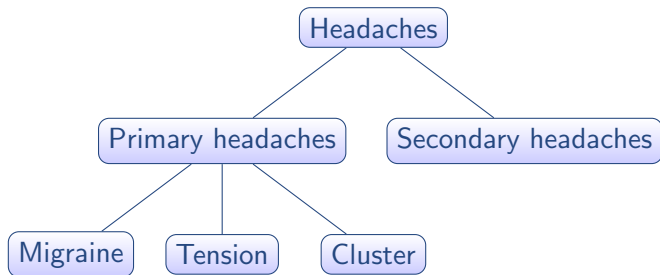
# Headaches



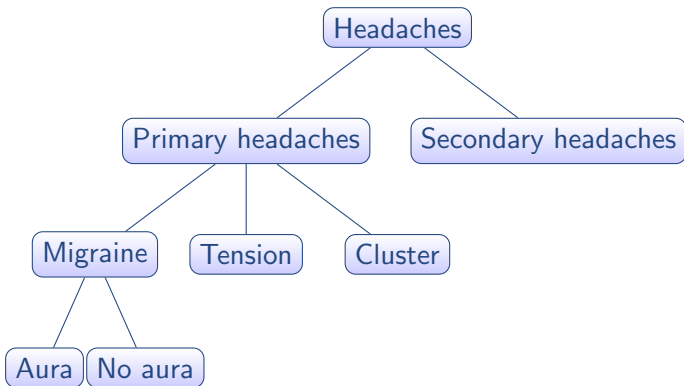
# Headaches



# Headaches



# Headaches

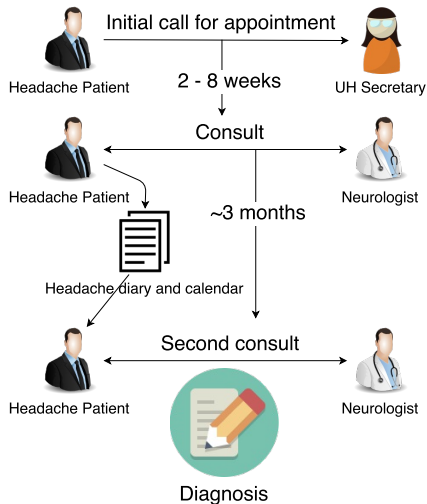




# Current process UH Ghent

Current process at UH Ghent is:

- ▶ Not digital
- ▶ cumbersome
- ▶ long-lasting



So there is need for a better (digital) alternative! This alternative has to:

- ▶ capture at least the same information as current solution
- ▶ be more efficient
- ▶ provide a second opinion for the doctors (auto-diagnose)

Intro

Machine learning

Platform requirements

Doctor dashboard

Mobile application

Conclusion & future work

Backend and data exposure

# Platform requirements

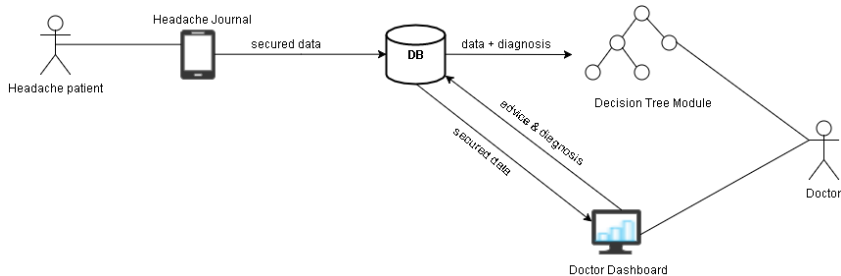
Our proposed alternative consists of:

- ▶ Headache journal: mobile app
- ▶ Doctor Dashboard: web application
- ▶ Machine learning module: decision support

Solution non-functional requirements:

- ▶ Security
- ▶ Availability
- ▶ Performance & learning curve
- ▶ Usability

# Platform requirements



Intro

Machine learning

Platform requirements

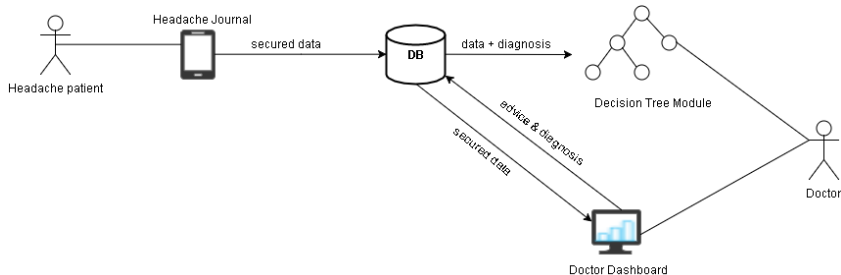
Doctor dashboard

Mobile application  
Chronicals

Conclusion & future work

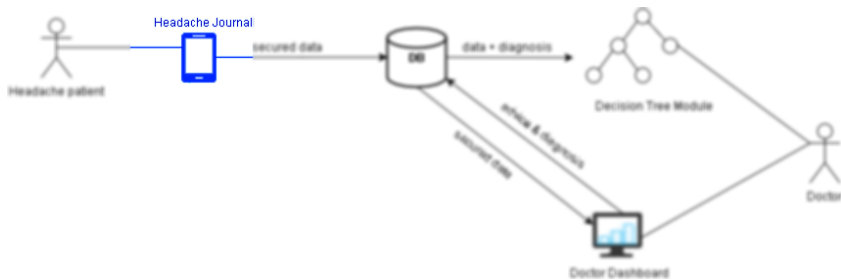
Backend and data exposure

# Mobile Application





# Mobile Application



# Mobile Application

Why create a new application?

## Competition

- ▶ Migraine Buddy
- ▶ Headache Diary
- ▶ Pfizer headache journal

# Mobile Application

Why create a new application?

## Competition

- ▶ Migraine Buddy
- ▶ Headache Diary
- ▶ Pfizer headache journal

All good, but:

# Mobile Application

Why create a new application?

## Competition

- ▶ Migraine Buddy
- ▶ Headache Diary
- ▶ Pfizer headache journal

All good, but:

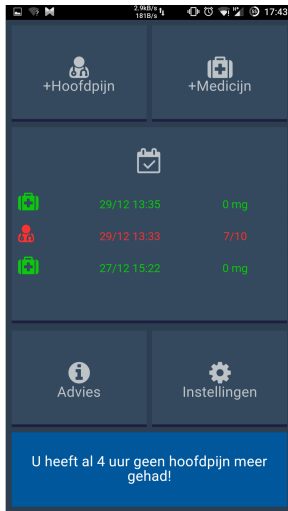
- ▶ none captures all data needed
- ▶ none offers usable data export

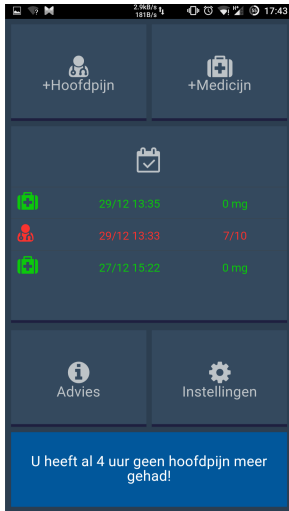
## Cross platform vs Native

	Native	Cross-platform
+	+ Native UX	+ 1 language
	+ device-specific features	+ Write once, run everywhere
	+ Better performance	+ Less maintenance
-	- Multiple languages	- Slower (lower performance)
	- Time consuming (development)	- Less device specific features
		- Harder to release online (Play Store/App Store)

## Cross platform vs Native

	Native	Cross-platform
+	<ul style="list-style-type: none"> <li>+ Native UX</li> <li>+ device-specific features</li> <li>+ Better performance</li> </ul>	<ul style="list-style-type: none"> <li>+ 1 language</li> <li>+ Write once, run everywhere</li> <li>+ Less maintenance</li> </ul>
-	<ul style="list-style-type: none"> <li>- Multiple languages</li> <li>- Time consuming (development)</li> </ul>	<ul style="list-style-type: none"> <li>- Slower (lower performance)</li> <li>- Less device specific features</li> <li>- Harder to release online (Play Store/App Store)</li> </ul>





# Chronicals

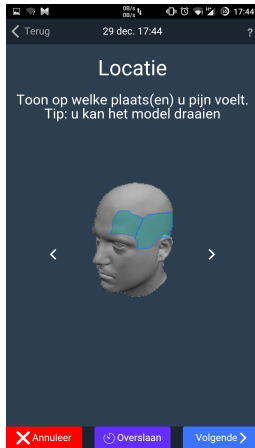




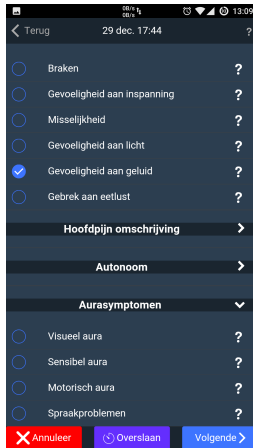
# Chronicals



# Chronicals

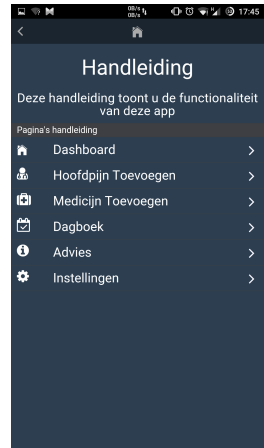
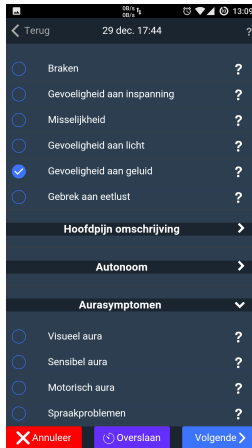
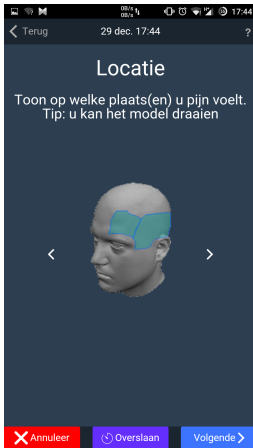


Mobile application



Chronicals

# Chronicals



Intro

Machine learning

Platform requirements

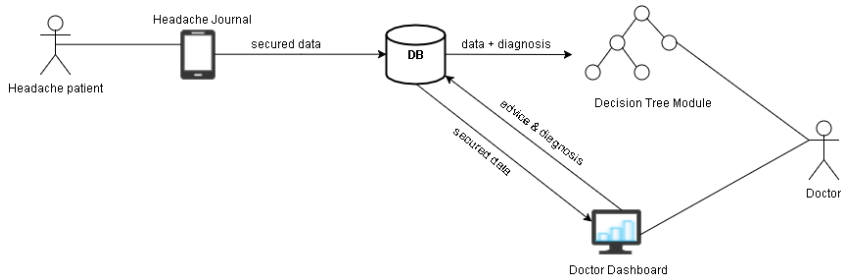
Doctor dashboard

Mobile application

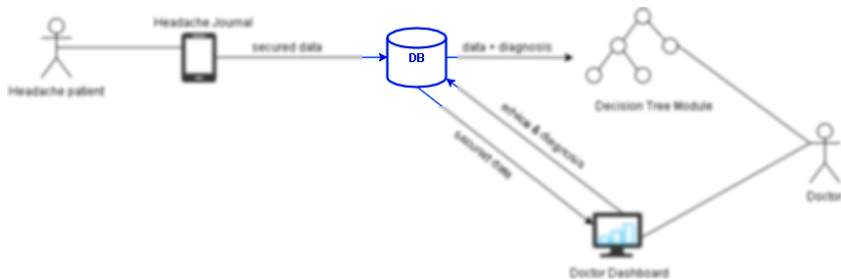
Conclusion & future work

Backend and data exposure

## Backend and data exposure



## Backend and data exposure

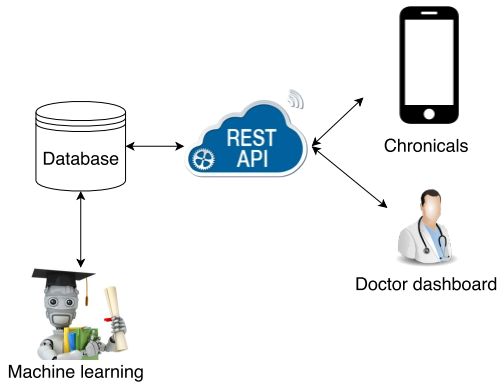


# Backend and data exposure

## Components

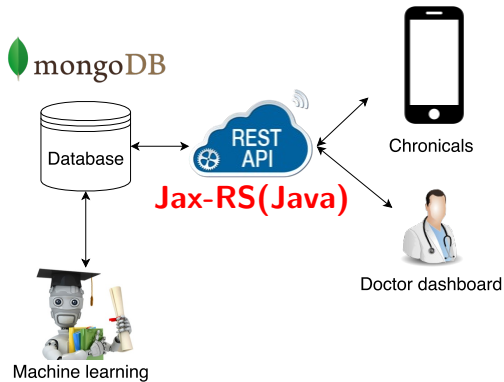
- ▶ Database
- ▶ Connection to App
- ▶ Connection to Doctor Dashboard
- ▶ Connection Machine learning module

# System





# System



Intro

Platform requirements

Mobile application

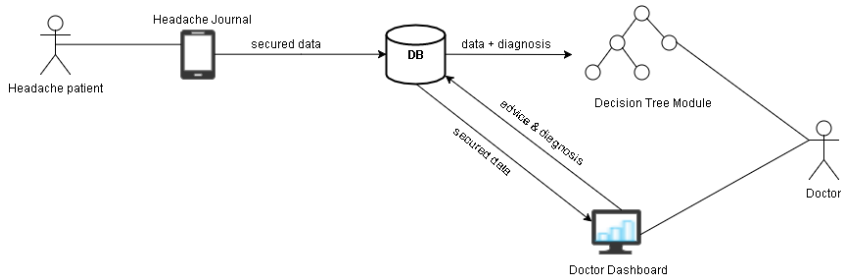
Backend and data exposure

Machine learning

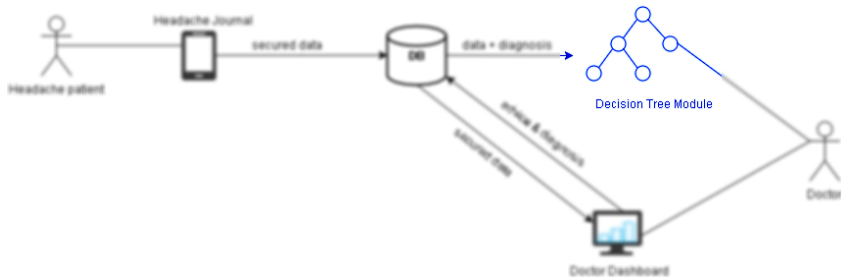
Doctor dashboard

Conclusion & future work

# Machine learning



# Machine learning



# Machine Learning

Decision support ( $\neq$  decision making)  $\Rightarrow$  White box model

## Possible models

- ▶ Decision trees
- ▶ Random Forests (Gray box)
- ▶ Bayesian networks

# Machine Learning

Decision support ( $\neq$  decision making)  $\Rightarrow$  White box model

## Possible models

- ▶ Decision trees
- ▶ Random Forests (Gray box)
- ▶ Bayesian networks

## Genetic merging of DT's



C4.5 (C5.0)



CART



QUEST

...

→ **Which tree is the most beautiful?**

# Current ensembles lack interpretability

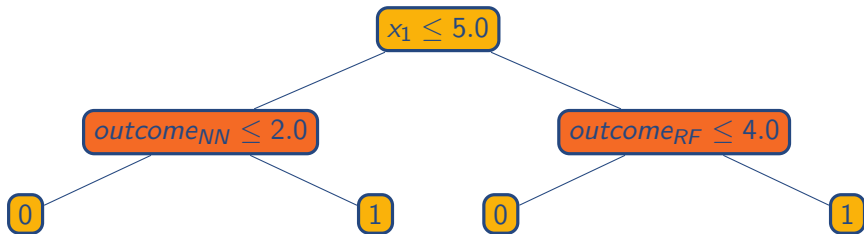
**Boosting, bagging, random forests, etc.** require majority voting (classification) or mean calculation (regression) to obtain prediction





## Current ensembles lack interpretability

The final decision tree obtained by **stacking** contains uninterpretable internal nodes

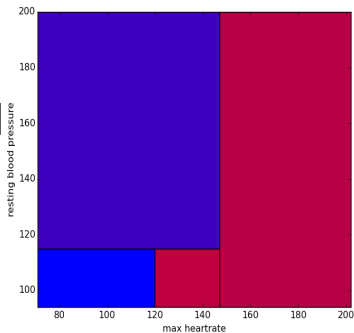
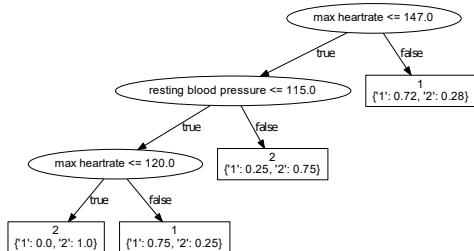


## Decision tree $\rightarrow$ decision space

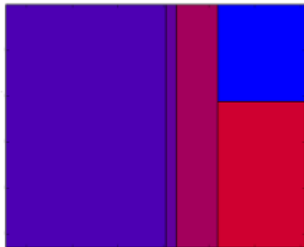
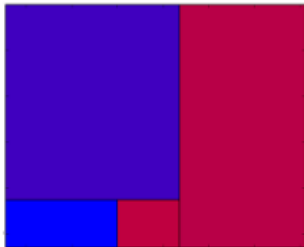
### Converting decision trees to decision spaces

We can define a one-to-one mapping between a decision tree and a set of  $k$ -dimensional hyperplanes ( $k = \text{\#features}$ ), called **decision space**. Each node in the decision tree corresponds to a hyperplane in the decision space.

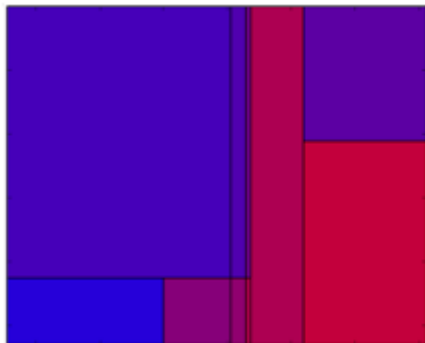
## Decision tree → decision space



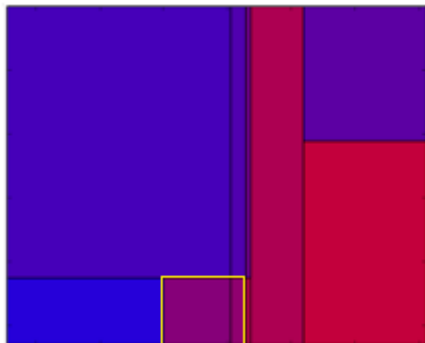
# Merging decision spaces



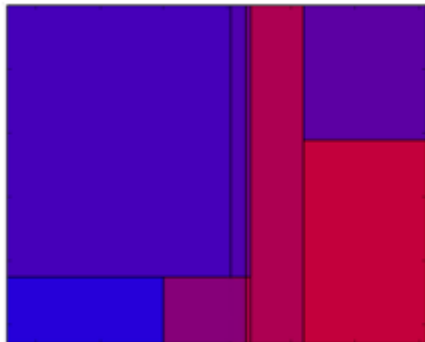
# Merging decision spaces



# Pruning decision spaces



# Pruning decision spaces



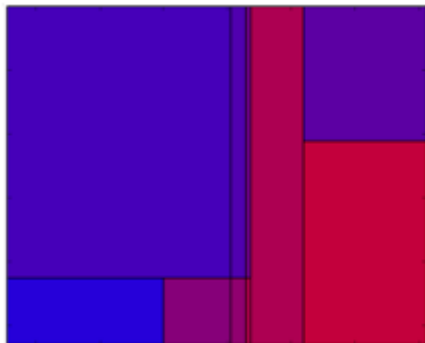
## Decision space $\rightarrow$ decision tree

### Converting decision spaces to decision trees

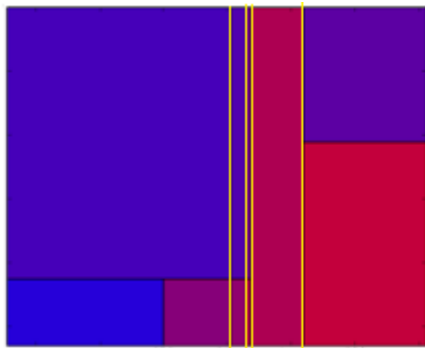
One-to-one mapping from decision tree to space is lost during conversion because the order is lost. Therefore, a **heuristic** approach must be taken, identifying **hyperplane candidates** and calculating a metric to choose the 'best' plane.



## Decision space $\rightarrow$ decision tree



## Decision space $\rightarrow$ decision tree

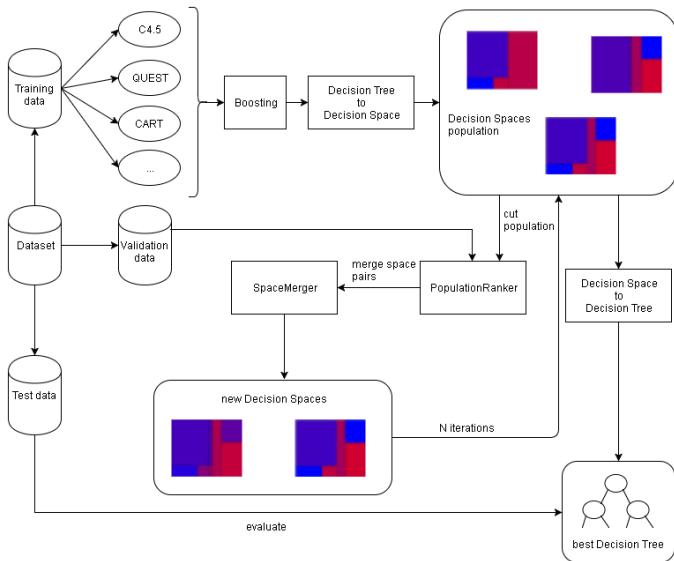


## Decision space $\rightarrow$ decision tree

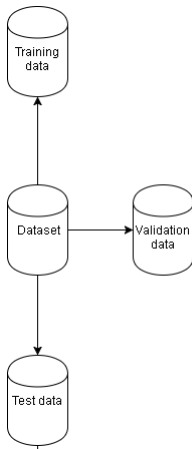
### Finding 'best' candidate hyperplane

Apply metric function to each plane, these include:

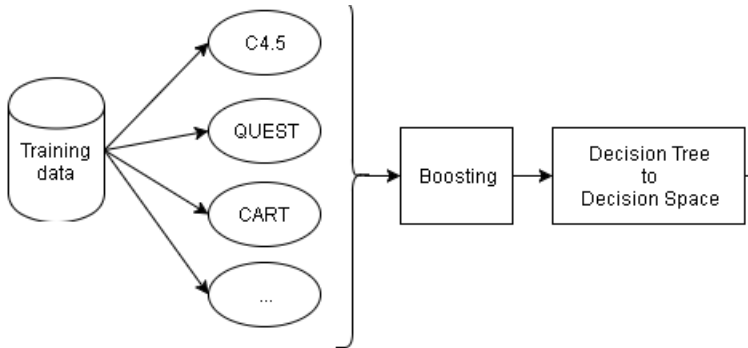
- ▶ information gain and Gini
- ▶ pick plane from most correlated feature
- ▶ pick plane that divide space in two most equal subspaces
- ▶ combination



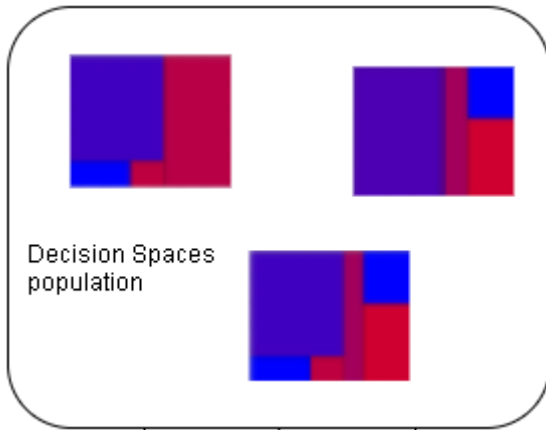
# Splitting the data



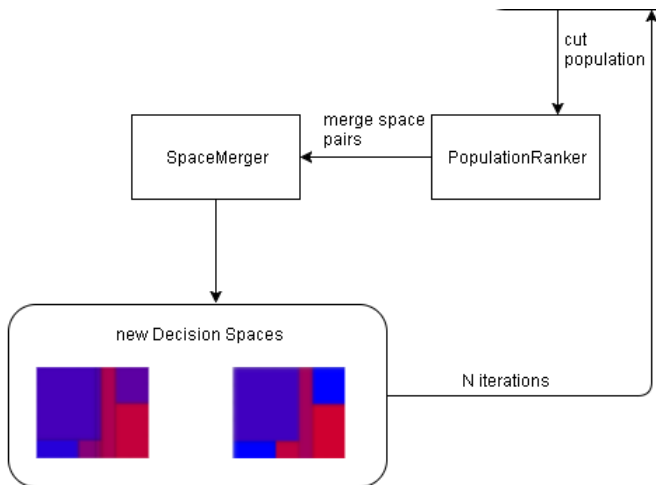
## Generate different decision trees



## Generate different decision trees



## Genetic merging



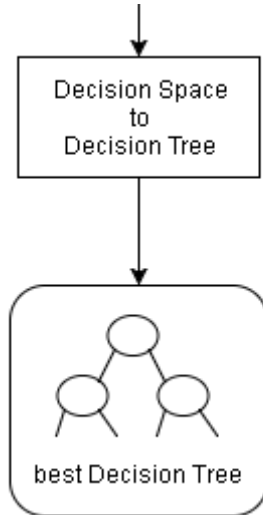


# PopulationRanker

## Fitness function

A high accuracy is the most important property of a decision tree, followed by its' size ( $\rightarrow$  comprehensibility). Genetic algorithms are well suited for **multi-objective optimization**.

## Final iteration

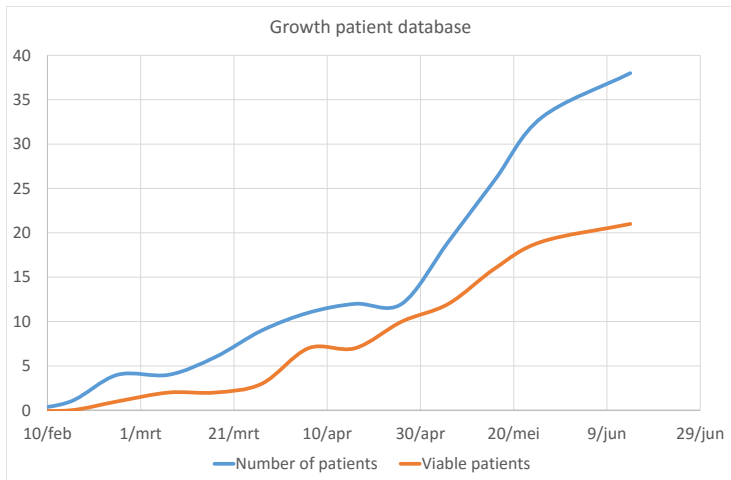


# Headache dataset

Data collection could only start in March:

- ▶ the mobile application had to be finished first
- ▶ an ethical committee had to approve our application

→ too few samples for machine learning



## Evaluating our algorithm

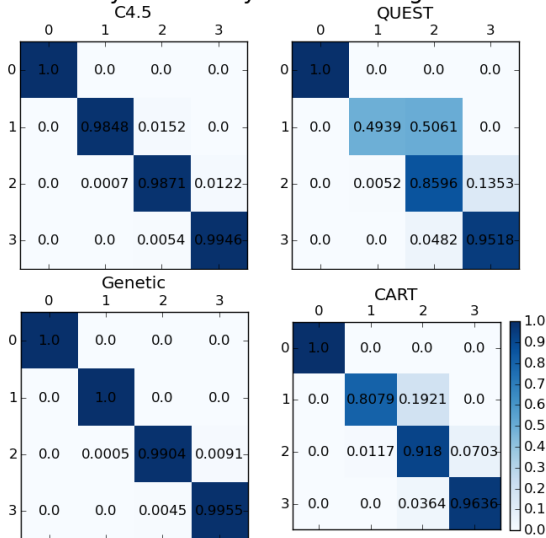
5 datasets from UCI

optimal parameters, feature selection when needed and k-fold CV

Name	#Samples	#Disc	#Cont	#Class	Imbalance rate
Heart	270	7	6	2	0.058
Car	1728	6	0	4	0.225
Iris	150	0	4	3	0
Shuttle	14500	0	9	7	0.18308
Nursery	12960	8	0	5	0.1498

Dataset	Folds	C4.5	CART	QUEST	Genetic
Heart disease	5	<u><b>0.8067</b></u>	0.7844	0.7844	<u><b>0.8067</b></u>
	10	<u><b>0.8104</b></u>	0.7732	0.7881	0.7993
Iris	3	0.9533	0.9467	0.9467	<u><b>0.96</b></u>
	5	0.9467	0.9333	0.9467	<u><b>0.9533</b></u>
Cars	3	<u><b>0.9722</b></u>	0.9693	0.9229	0.9693
	5	0.9711	0.9682	0.9241	<u><b>0.9786</b></u>
	10	0.9756	0.9751	0.9265	<u><b>0.9803</b></u>
Shuttle	3	0.9987	0.9983	0.9964	<u><b>0.9988</b></u>
	5	0.9986	0.9981	0.9962	<u><b>0.9988</b></u>
	10	0.9990	0.9987	0.9941	<u><b>0.9992</b></u>
Nursery	3	0.9890	0.9431	0.9147	<u><b>0.9914</b></u>
	5	0.9918	0.9498	0.9251	<u><b>0.9958</b></u>
	10	0.9937	0.9568	0.9259	<u><b>0.9954</b></u>

## Accuracy on nursery dataset using 10 folds



Intro

Machine learning

Platform requirements

Doctor dashboard

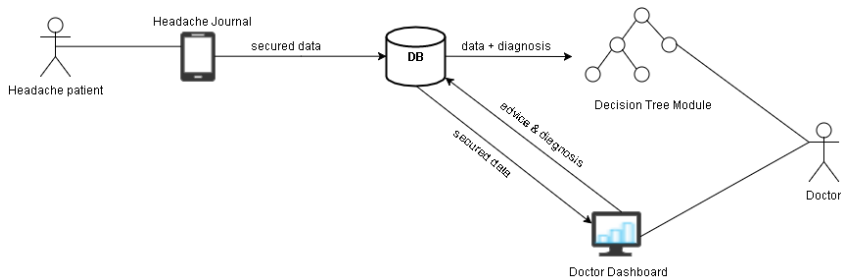
Mobile application

Conclusion & future work

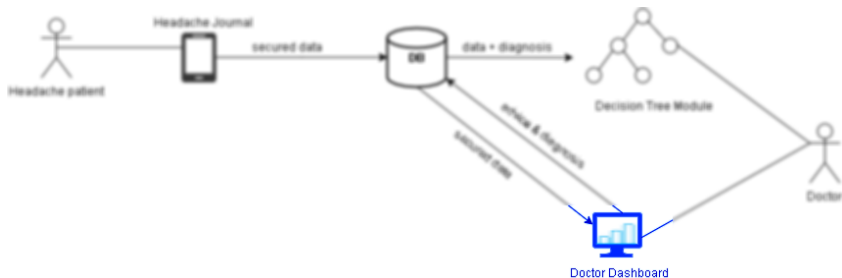
Backend and data exposure



## Doctor dashboard



# Doctor dashboard



# Doctor Dashboard

- ▶ Web application in order for the doctors to access the data exposed by our REST API
- ▶ Preferably in the form of visualizations, which allow to process a lot of data in a small amount of time
- ▶ Developed by Maarten Vanden Berghe

Intro

Machine learning

Platform requirements

Doctor dashboard

Mobile application

Conclusion & future work

Backend and data exposure

## Conclusion

- ▶ It is shown that the current process in the UH of Ghent can be completely digitized. Our solution can significantly improve the efficiency and reduce the frequency of consults, leading to a reduction in health care costs.
- ▶ The foundations for a diagnosis support system are built, using a genetic approach to merge different induced decision trees to obtain a single decision tree with enhanced accuracy.

## Future work

- ▶ Develop native applications for iOS and Android to enhance look-&-feel
- ▶ Re-evaluate our machine learning models on a larger headache dataset
- ▶ Implement more induction algorithms and ensemble techniques to create a more diverse initial population
- ▶ Experiment with other selection techniques and fitness functions
- ▶ Optimize the heuristic approach to convert decision spaces to decision trees

# Thank you for your attention!

Intro

Platform requirements

Mobile application

Backend and data exposure

Machine learning

Doctor dashboard

Conclusion & future work