

May - June 2020

Saber PRO success prediction model using decision tree based learning

Gregorio Pérez Bernal
Luisa Toro Villegas

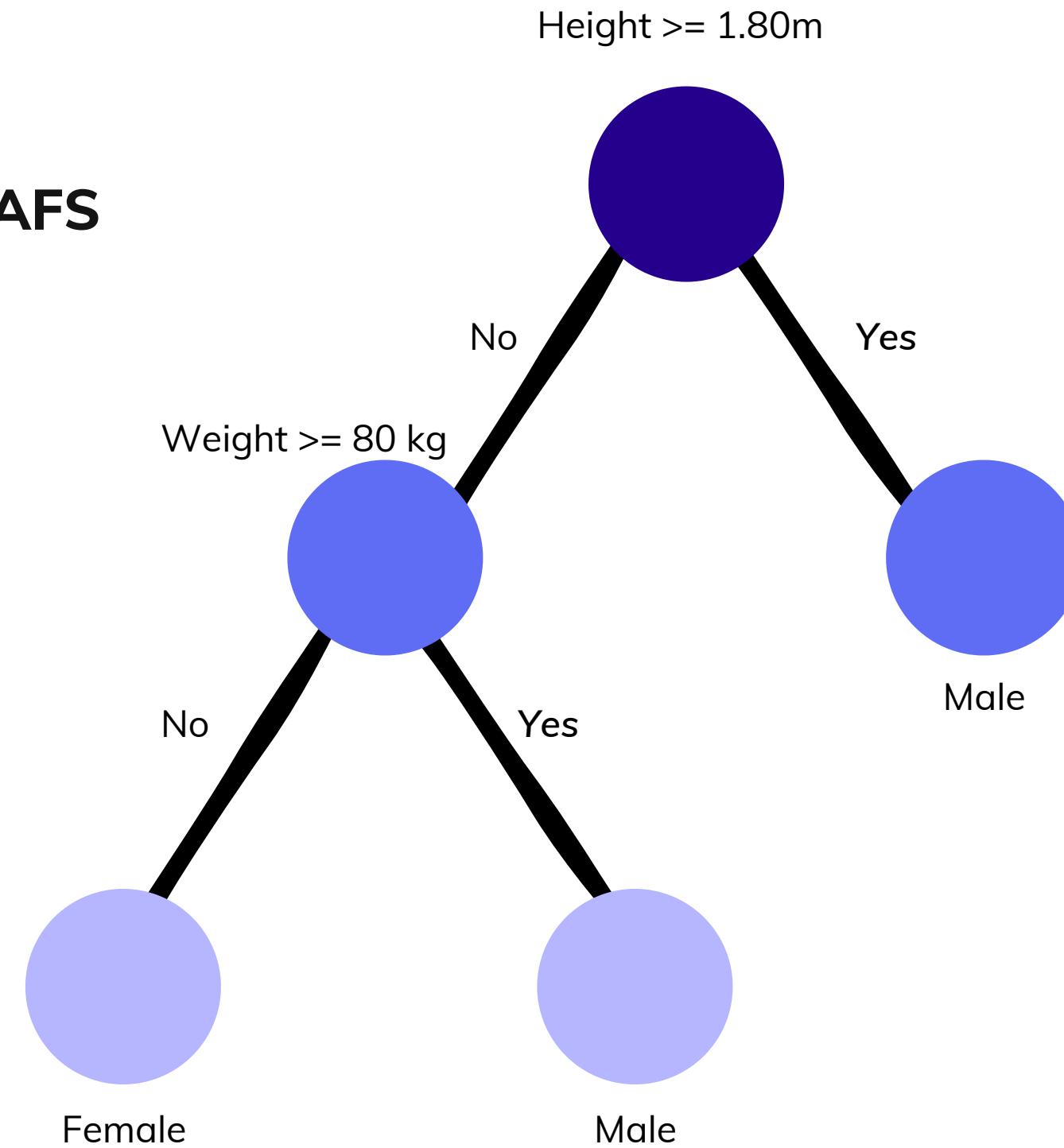
CART Algorithm

CLASSIFICATION AND REGRESSION TREE

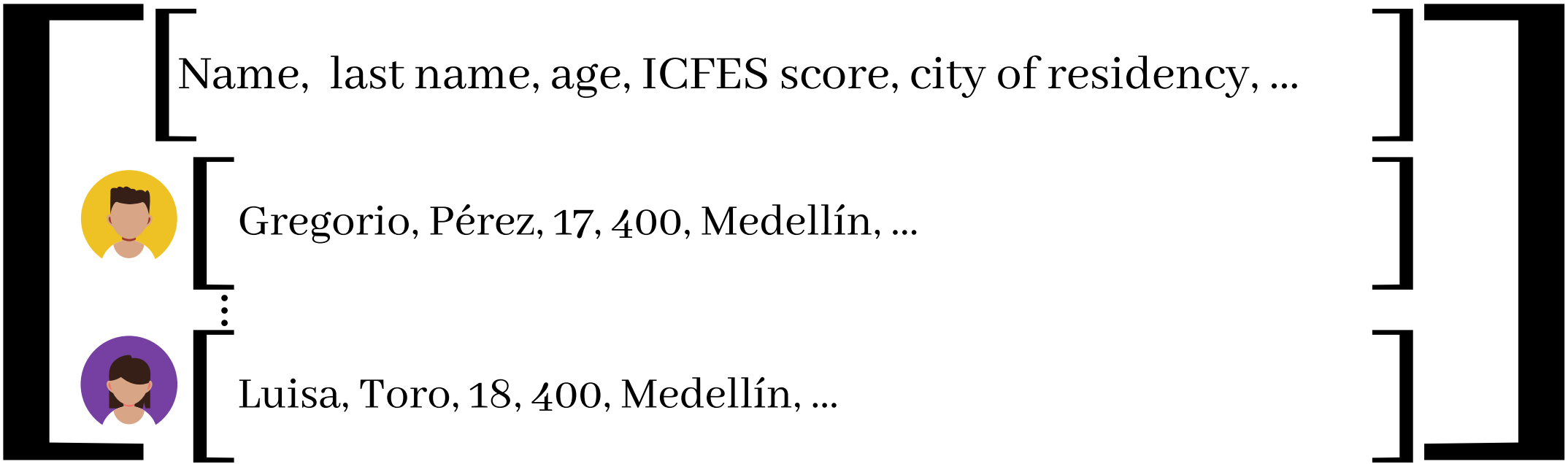
◆ BINARY TREE WITH DECISION NODES AND LEAFS

◆ GINI IMPURITY AS DECISION CRITERIA

Greedy or binary recursive
decision



Data structure : array of arrays



Complexity per method

Method	Complexity
create Matrix	$O(n)$
addPerson	$O(1)$
Access	$O(1)$

*n = rows or number of subjects.

Complexity per method

Method	Complexity
importData	$O(n)$
Question (<code>_init_</code>)	$O(1)$
Question (match)	$O(1)$
Question (toString)	$O(1)$
Partition	$O(n)$
Decide partition	$O(n^2)$
Tree (<code>_init_</code>)	$O(2^m)$
Tree (generateString)	$O(1)$

is_number	$O(1)$
classCounts	$O(n)$
countSuccess	$O(n)$
bestValue	$O(n)$
gini	$O(1)$
informationGain	$O(1)$
prediction	$O(n)$
classify	$O(m)$
runClassify	$O(n*m)$

*n = rows or number of subjects.

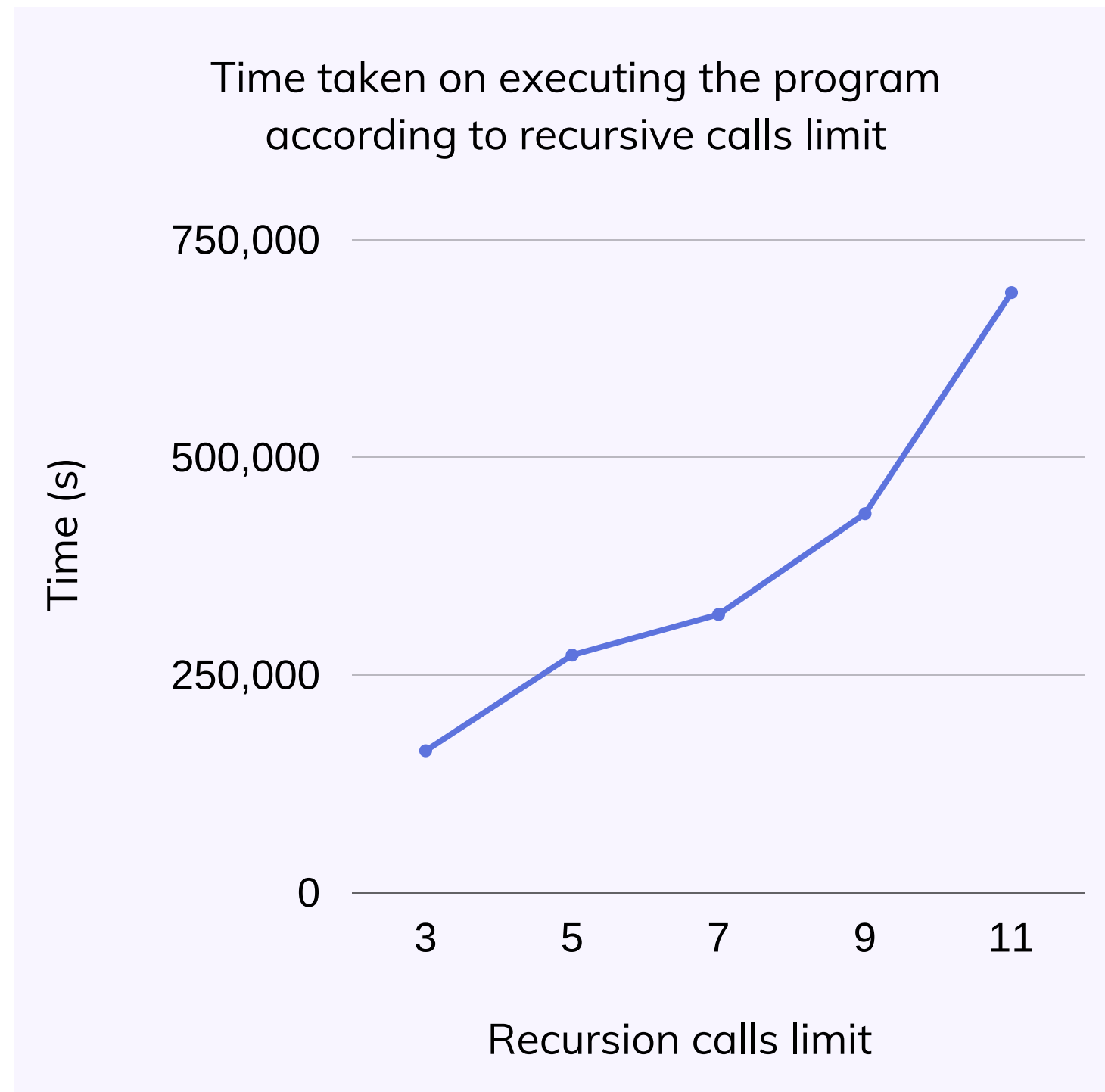
*m = columns or number of categories.

Time profile

Time taken executing each method according to recursive depth

	Average time per method (s)				
Recursion calls	Dividing dataset	Decide partition	Building tree	Classification	Total time (s)
3	0,121	10,670	160,915	0,516	163,306
5	0,049	4,273	270,709	0,919	273,126
7	0,190	1,419	316,317	1,670	319,757
9	0,005	0,458	433,652	2,413	437,897
11	0,003	0,244	689,195	2,541	693,612

*Trained with 135000 subjects and tested with 45000



*Trained with 135000 subjects

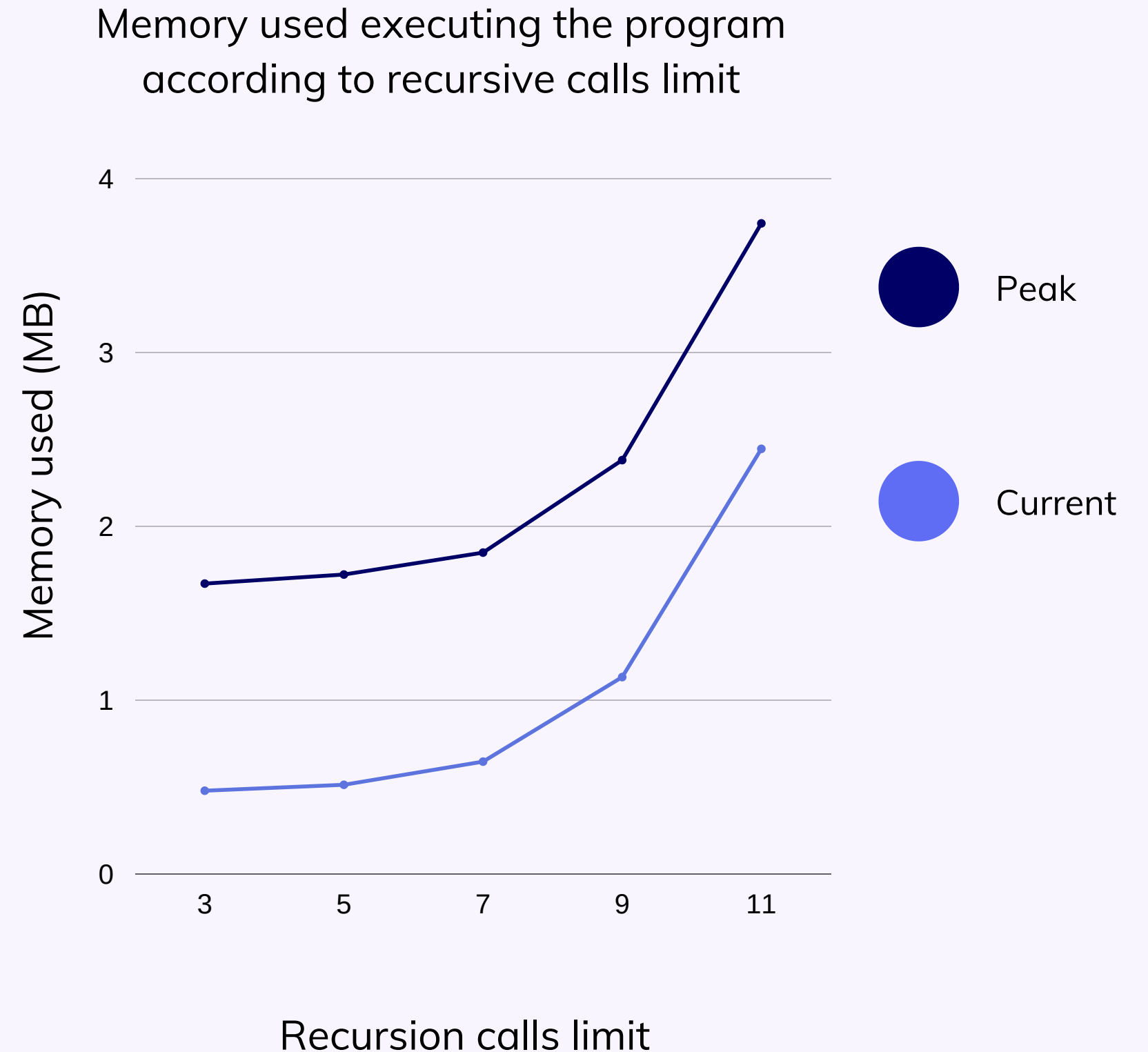
Memory profile

Memory complexity of the main data structures used

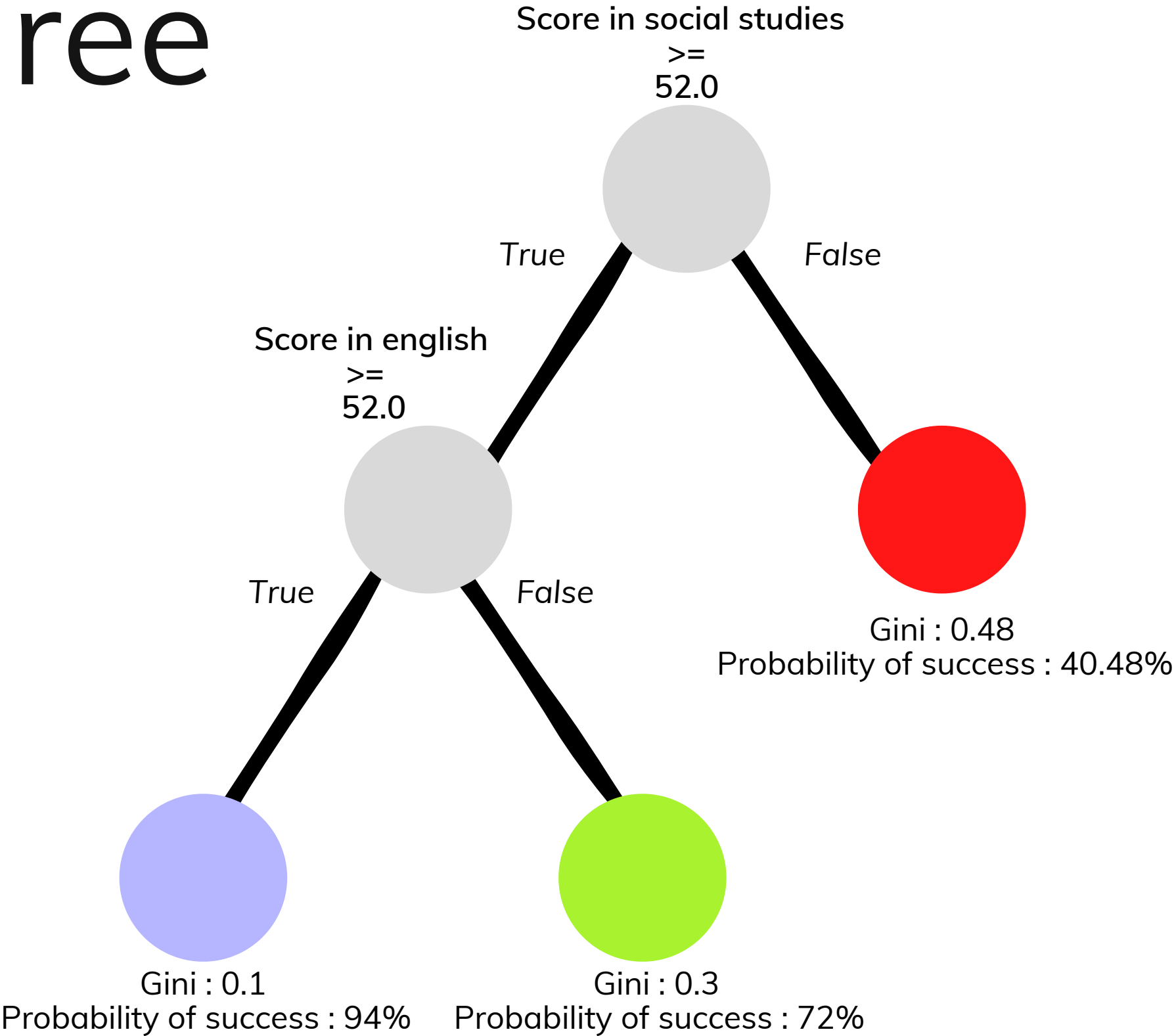
Data structure	Memory complexity
Tree	$O(2^m)$
Matrix	$O(n*m)$
Dictionary	$O(n)$

*n = rows or number of subjects.

*m = columns or number of categories.



Tree



Accuracy percentage based on recursion depth

Recursive calls	Accuracy
5	77.58%
6	78.16%
7	78.97%
8	79.42%
9	79.68 %
10	79.59%
11	79.80 %

*Trained with 135000 subjects
and tested with 45000

Confusion matrix

		Actual class	
		1	0
Predicted class	1	4615	4615
	0	4848	17867

*Trained with 135000 subjects and tested with 45000

*Recursion depth = 7

Possible applications

Using the algorithm in real life situations.



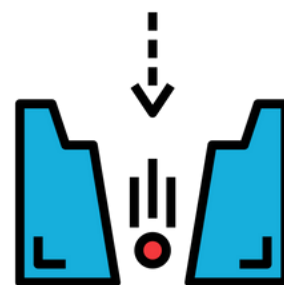
ASSIGNING SCHOLARSHIPS



UNIVERSITY ENTRY CRITERIA



DECIDING INVESTING SECTORS



BREACHING SOCIAL GAPS THAT AFFECT SUCCESS