

Ejercicio 1

El modelo de Random Forest fue probado con los parametros por defecto, y tambien cambiando el criterio de seleccion de atributos y el tamaño del subconjunto de atributos sobre el cual se realiza el split. Los resultados fueron siempre los mismos y son:

accuracy: 95.56%

ConfusionMatrix:

True:	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa:	15	0	0
Iris-versicolor:	0	15	2
Iris-virginica:	0	0	13

kappa: 0.933

ConfusionMatrix:

True:	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa:	15	0	0
Iris-versicolor:	0	15	2
Iris-virginica:	0	0	13

Ejercicio Cardiac

Performance del modelo decision Tree simple (parametros defecto):

accuracy: 95.12%

ConfusionMatrix:

True:	Si	No
Si:	20	2
No:	0	19

precision: 100.00% (positive class: No)

ConfusionMatrix:

True:	Si	No
Si:	20	2
No:	0	19

recall: 90.48% (positive class: No)

ConfusionMatrix:

True:	Si	No
Si:	20	2
No:	0	19

AUC (optimistic): 0.988 (positive class: No)

AUC: 0.970 (positive class: No)

AUC (pessimistic): 0.952 (positive class: No)

Performance del modelo Random Forest con los mismos parametros:

accuracy: 95.12%

ConfusionMatrix:

True:	Si	No
Si:	19	1
No:	1	20

precision: 95.24% (positive class: No)

ConfusionMatrix:

True:	Si	No
Si:	19	1
No:	1	20

recall: 95.24% (positive class: No)

ConfusionMatrix:

True:	Si	No
Si:	19	1
No:	1	20

AUC (optimistic): 0.962 (positive class: No)

AUC: 0.962 (positive class: No)

AUC (pessimistic): 0.962 (positive class: No)

Se utilizó el operador optimize parameters para encontrar los parámetros óptimos para el Random Forest. Los parámetros del optimizador son los parámetros por defecto, que son:

Optimize Parameters (Evolutionary)

Edit Parameter Settings...

error handling: fail on error

max generations: 50

☐ use early stopping

☒ specify population size

population size: 5

☒ keep best

mutation type: gaussian_mutation

selection type: tournament

tournament fraction: 0.25

crossover prob: 0.9

☐ use local random seed

☐ show convergence plot

Se configuró este operador para que encuentre los subset_ratio y number_of_trees óptimos, y los resultados fueron los siguientes:

PerformanceVector [
-----accuracy: 97.56%

ConfusionMatrix:

True:	Si	No
Si:	20	1
No:	0	20

-----precision: 100.00% (positive class: No)

ConfusionMatrix:

True:	Si	No
Si:	20	1
No:	0	20

-----recall: 95.24% (positive class: No)

ConfusionMatrix:

True:	Si	No
Si:	20	1
No:	0	20

-----AUC (optimistic): 0.993 (positive class: No)

-----AUC: 0.989 (positive class: No)

-----AUC (pessimistic): 0.986 (positive class: No)

]

Random Forest (2).subset_ratio = 0.5646206387581426

Random Forest (2).number_of_trees = 8