Criterios de corrección de la práctica 1

Aprendizaje Automático

En este documento se especifican los criterios que se seguirán para corregir los ejercicios correspondientes a la primera práctica. En estos ejercicios, se pide el desarrollo de una serie de funciones que tienen una puntuación distinta en la nota. En la siguiente tabla se puede ver la puntuación de cada una de las funciones a realizar:

	Nombre de la función	Puntuación
Ejercicio 2	oneHotEncoding oscar	0.02
	calculateMinMaxNormalizationParameters oscar	0.02
	calculateZeroMeanNormalizationParameters oscar	0.02
	normalizeMinMax! oscar	0.02
	normalizeMinMax oscar	0.02
	normalizeZeroMean! oscar	0.02
	normalizeZeroMean oscar	0.02
	classifyOutputs hugo	0.25
	accuracy hugo	0.2
	buildClassANN iago	0.25
Ejercicio 3	holdOut hugo	0.1
	trainClassANN iago	0.32
Ejercicio 4	confusionMatrix (clasificación binaria) oscar	0.32
	confusionMatrix (clasificación multiclase) oscar	0.32
Ejercicio 5	crossvalidation hugo	0.1
	ANNCrossvalidation	0.3
Ejercicio 6	modelCrossValidation	0.2

Para conseguir la puntuación en cada función, deberá de desarrollarse correctamente, siguiendo la firma de la función especificada, devolviendo los valores descritos en el ejercicio, y siguiendo el resto de restricciones especificadas, como el número de bucles permitido, que también se deberán

cumplir. Es importante tener en cuenta que las funciones deberán haber sido definidas **exactamente** con las firmas especificadas. Además, para obtener la puntuación será necesario definir correctamente según estos criterios todas y cada una de las versiones de cada función especificadas en los ejercicios, en los casos en las que la función esté sobrecargada. La única excepción a esta regla está en la función *confusionMatrix*, que se evaluará de forma diferente para la clasificación binaria y multiclase.

En el caso de la función *trainClassANN*, esta se desarrolla en dos ejercicios distintos. Esta función se evalúa con la última modificación que se solicita, es decir, dentro del ejercicio 3.

Estas funciones se entregarán en un archivo con extensión .jl a través de Moodle. A pesar de haber sido realizada esta práctica por varias personas, solamente se realizará una entrega (sólo un miembro del equipo realizará la entrega), y el archivo tendrá el nombre indicado en la entrega. Este archivo solamente contendrá las funciones especificadas. Si se quieren hacer experimentos con estas funciones, lo más sencillo es crear un script de Julia aparte, que cargue este archivo de funciones mediante *include*, y realizar en él todas las operaciones que se desee. Para que sea más sencillo, se provee de un archivo llamado “firmas.jl” con una plantilla de lo que es necesario desarrollar, y que puede ser modificado.

De cara a poder obtener la nota correspondiente, cada persona deberá realizar una defensa individual de las funciones entregadas. Para algunas de ellas existen implementaciones ya realizadas y disponibles en distintas librerías. Como es obvio, es necesario desarrollar estas funciones sin hacer llamadas a las de esas librerías. En caso de detectarse en cualquier momento, incluida la defensa, que, en lugar de desarrollarla, se ha hecho una llamada a una función ya existente, esto invalidaría la nota de dicha función.

Finalmente, el archivo con las funciones entregadas será sometido a un análisis del sistema anti-plagio de la Universidad, y ante las coincidencias más allá de lo razonable se aplicará la normativa de la Universidad.