ugr | Universidad de Granada

undersampling

Una biblioteca en Scala para undersampling en clasificación no balanceada.

Alumno: Néstor Rodríguez Vico
Tutor 1: Alberto Fernández Hilario
Tutor 2: Salvador García López

25 de junio de 2018

Universidad de Granada

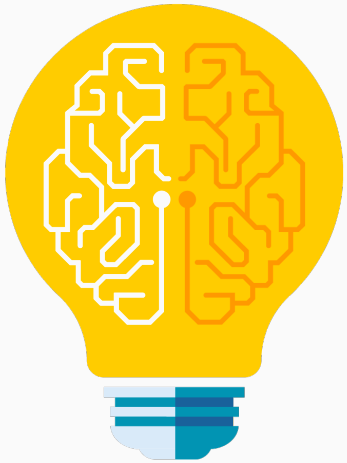
Índice.

Introducción.

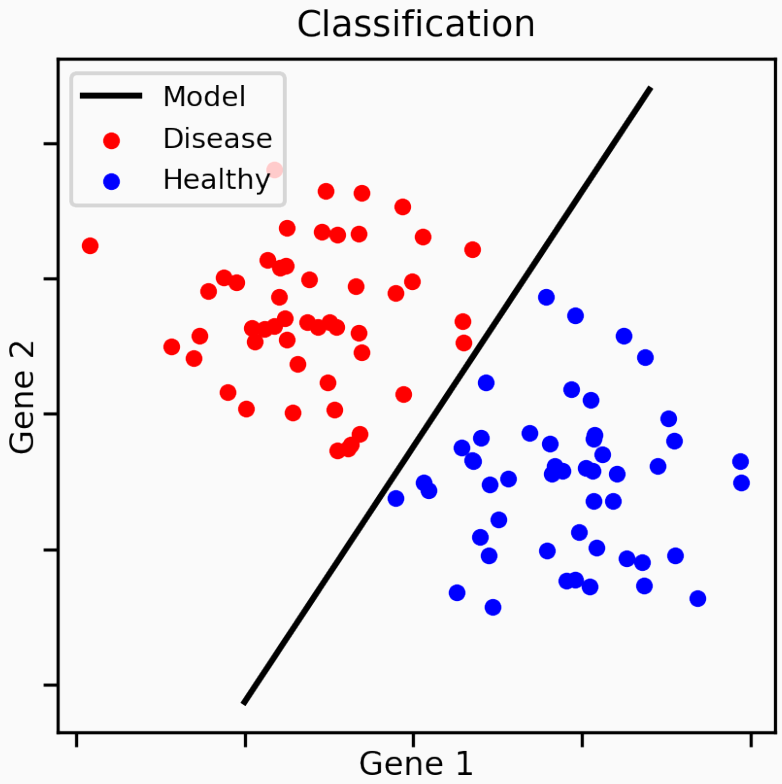
Repositorio.

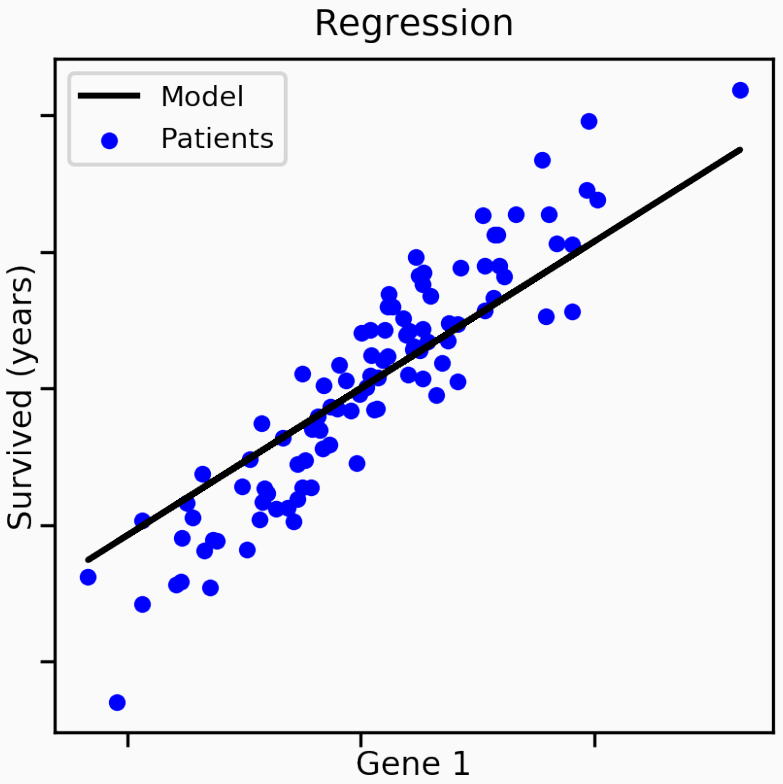
https://github.com/NestorRV/undersampling

Machine Learning.

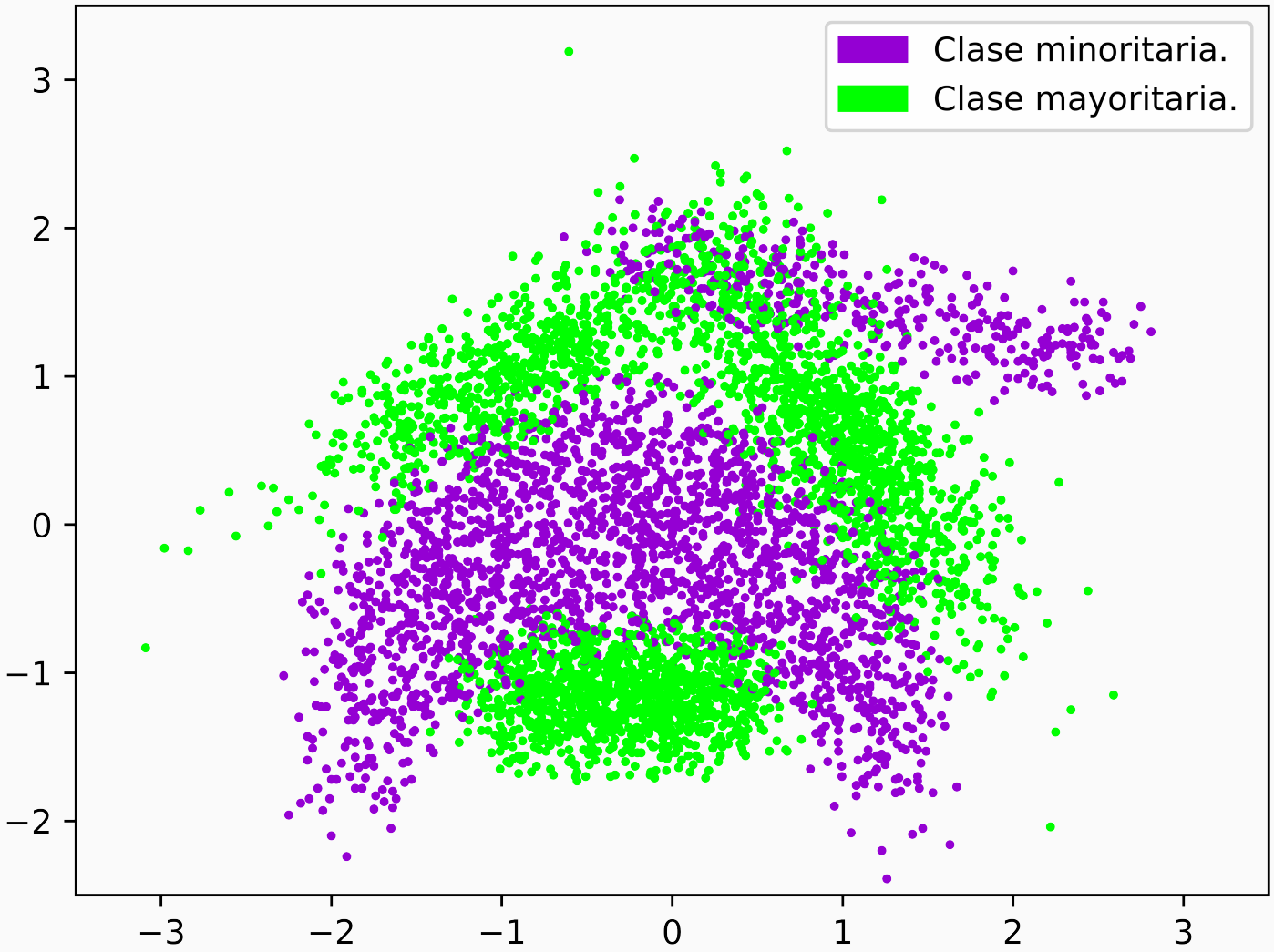


Clasificación vs. Regresión.

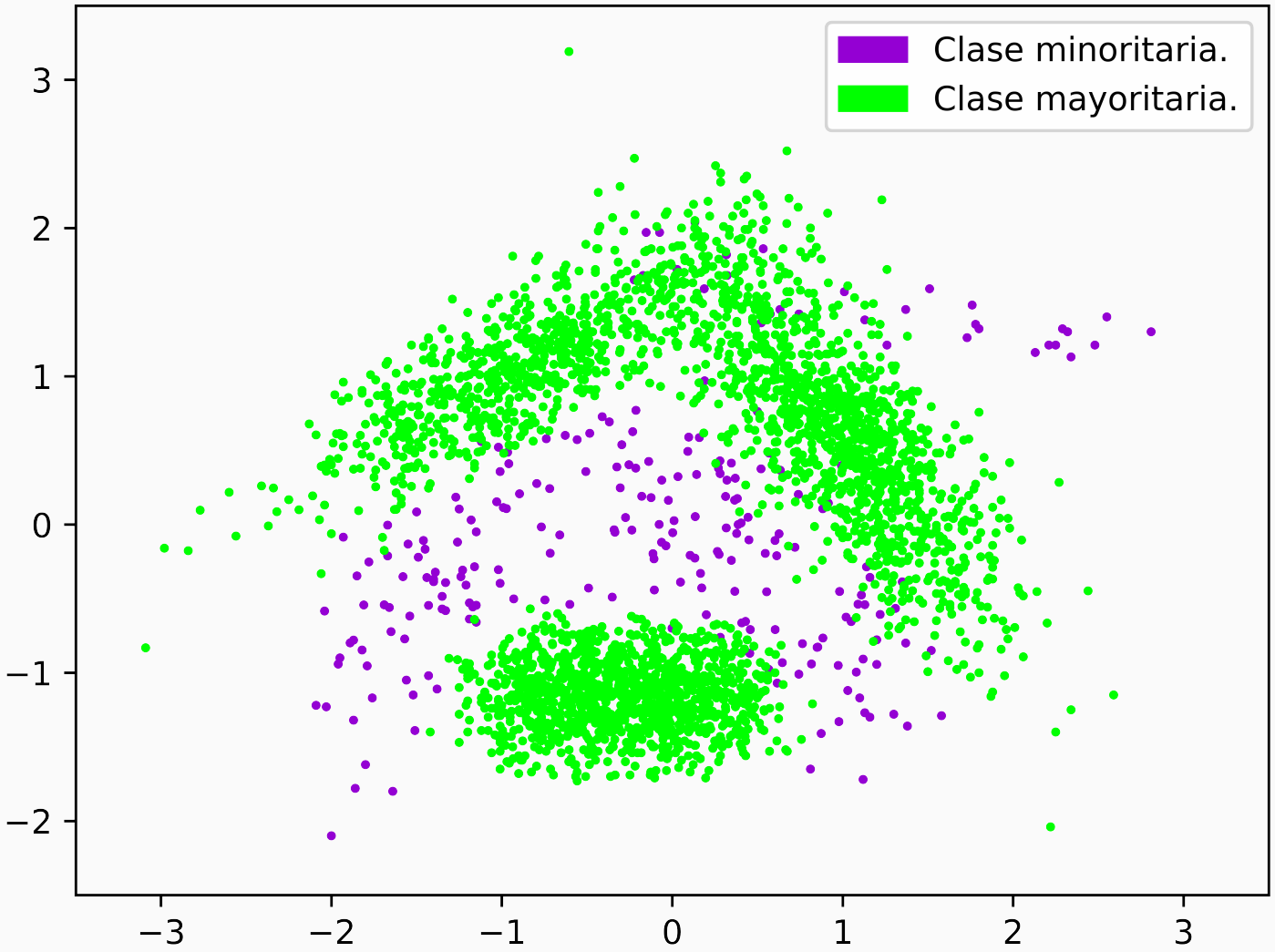




Clasificación no balanceada.



Clasificación no balanceada.



Objetivos.

Objetivos.

- Revisión bibliográfica del estado del arte.
- Estudio de requisitos y diseño de implementación.
- Implementación y prueba de los algoritmos.
- Comparación de los resultados obtenidos.

Planificación y diseño.

Planificación.

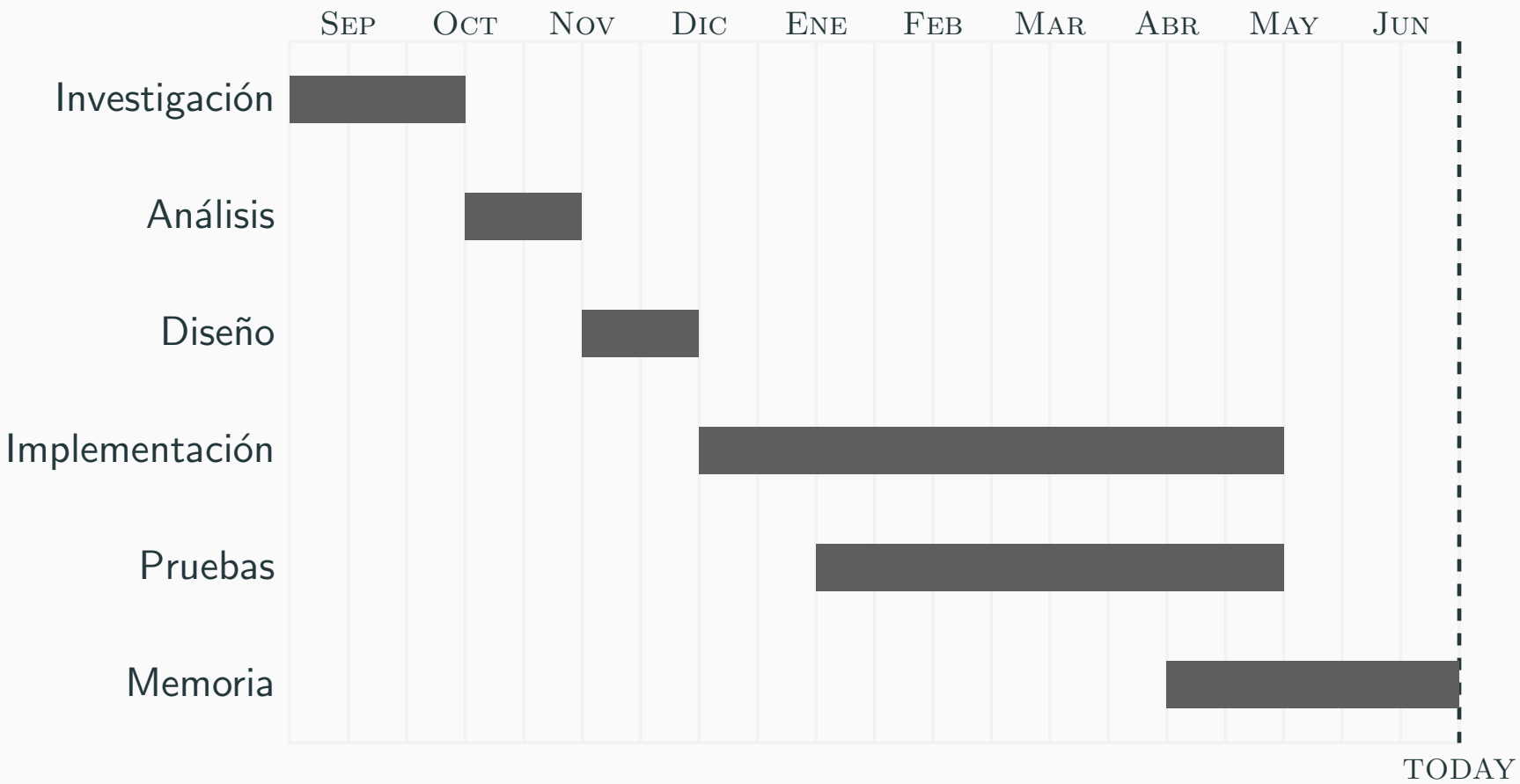


Diagrama de paquetes.

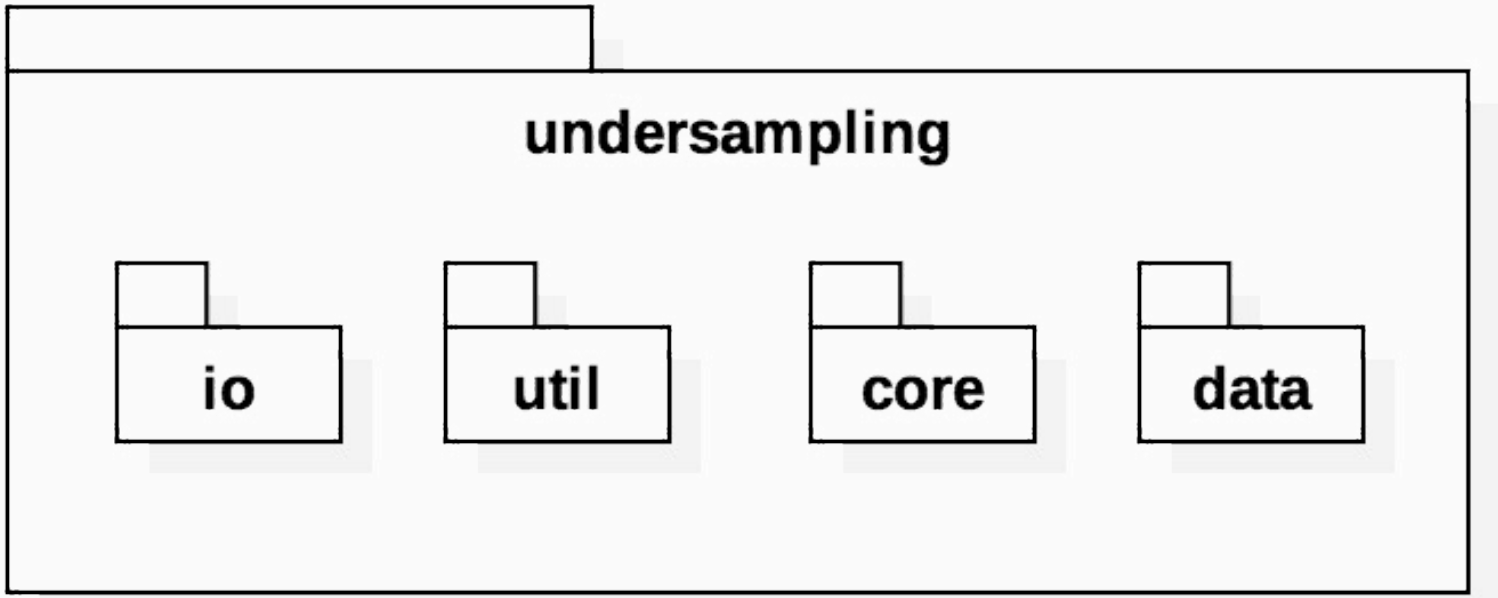
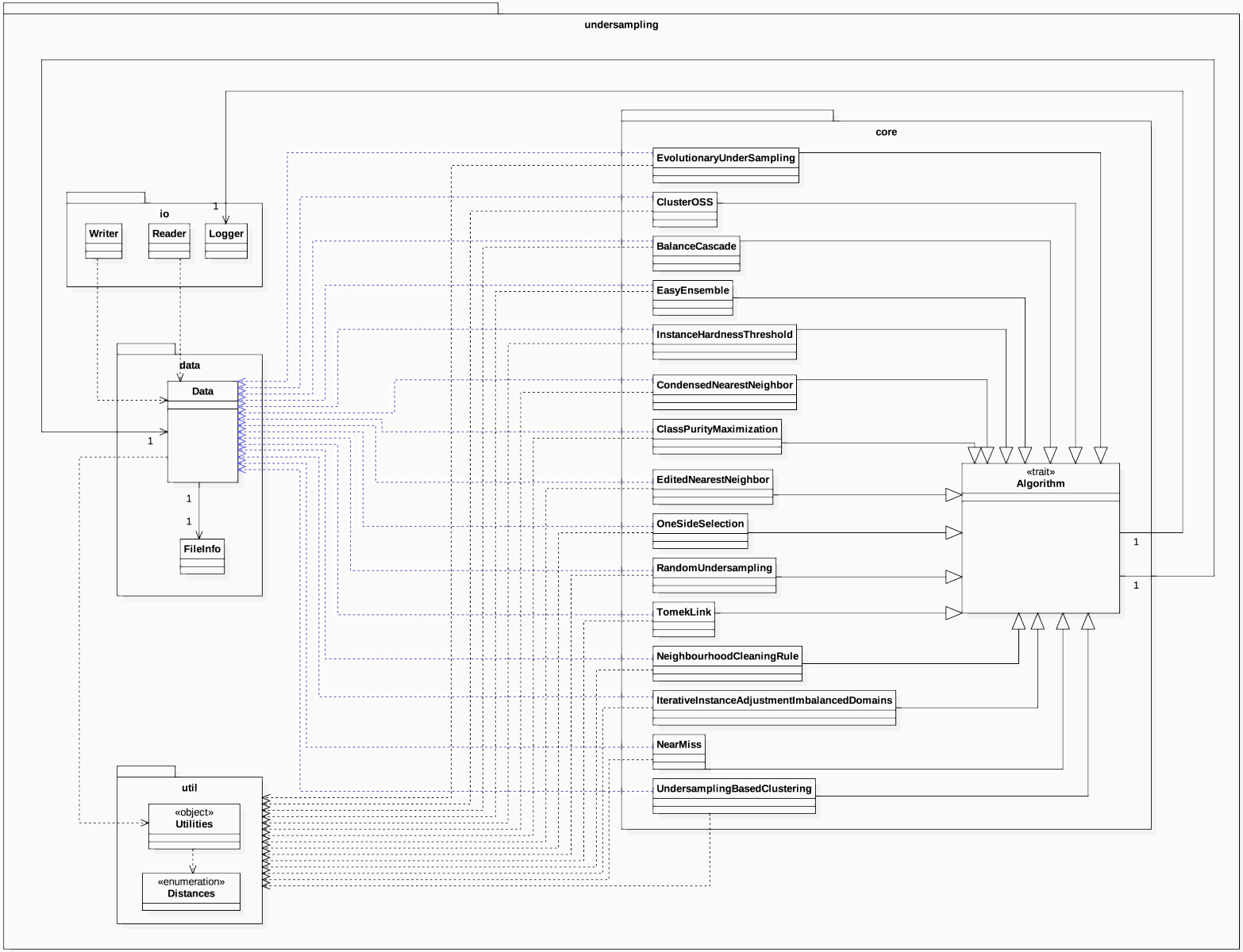


Diagrama de clases simplificado.



Requisitos funcionales.

- Alta velocidad de procesamiento.
- Diferentes formatos de entrada.
- Diferentes formatos de salida.
- Seguro a errores.
- Datos intactos.

Algoritmos seleccionados.

Balance Cascade.

"Exploratory Undersampling for Class-Imbalance Learning escrito por *Xu-Ying Liu, Jianxin Wu y Zhi-Hua Zhou"* [7]

Particiones $\rightarrow$ Clasificador $\rightarrow$ Combinar

Class Purity Maximization algorithm.

"An Unsupervised Learning Approach to Resolving the Data Imbalanced Issue in Supervised Learning Problems in Functional Genomics escrito por *Kihoon Yoon y Stephen Kwek"* [12]

Particiones en clústeres mientras obtengamos un conjunto más puro.

ClusterOSS.

"ClusterOSS: a new undersampling method for imbalanced learning." escrito por *Victor H Barella, Eduardo P Costa y André C. P. L. F. Carvalho* [1]

Clústeres $\rightarrow$ Elementos más representativos $\rightarrow$ Tomek Link

Condensed Nearest Neighbor decision rule.

"The Condensed Nearest Neighbor Rule" escrito por *P. Hart* [3]

Puntos innecesarios: se puede clasificar con el conjunto de puntos que tenemos actualmente.

Easy Ensemble.

"Exploratory Undersampling for Class-Imbalance Learning" escrito por *Xu-Ying Liu, Jianxin Wu y Zhi-Hua Zhou* [7]

Particiones $\rightarrow$ Clasificador $\rightarrow$ Combinar
Particiones $\rightarrow$ Combinar

Edited Nearest Neighbour rule.

"Asymptotic Properties of Nearest Neighbor Rules Using Edited Data" escrito por *Dennis L. Wilson* [5]

Puntos redundantes: podemos predecir su clase usando todos los puntos del conjunto original menos el.

Evolutionary Undersampling.

"Evolutionary Under-Sampling for Classification with Imbalanced Data Sets: Proposals y Taxonomy" escrito por *Salvador Garcia y Francisco Herrera* [2]

Vector de tamaño $|conjunto\_original|$ formado por unos y/o ceros: representa un subconjunto de elementos del conjunto original. Si en la posición *i* del vector hay un uno, la instancia *i* está incluida en dicho subconjunto.

Instance Hardness Threshold.

"An Empirical Study of Instance Hardness" escrito por *Michael R. Smith, Tony Martinez y Christophe Giraud-Carrier* [9]

Clasificar las instancias según su dureza: alta, baja o ninguna. Las instancias con una dureza alta se consideran que están mal clasificadas o son ruidosas $\rightarrow$ son eliminadas del conjunto de datos.

Iterative Instance Adjustment for Imbalanced Domains.

"Addressing imbalanced classification with instance generation techniques: IPADE-ID" escrito por *Victoria López, Isaac Triguero, Cristóbal J. Carmona, Salvador García y Francisco Herrera* [8]

Inicializar la población con las instancias más relevantes $\rightarrow$ Aplicamos de forma iterativa una versión simplificada del algoritmo *Differential Evolution*.

NearMiss.

"kNN Approach to Unbalanced Data Distribution: A Case Study involving Information Extraction" escrito por *Jianping Zhang y Inderjeet Mani* [13]

Eliminar los vecinos mayoritarios que sean redundantes:

NearMiss 1: Distancia media a sus tres vecinos de la clase minoritaria más cercanos sea menor.
NearMiss 2: Distancia media a sus tres vecinos de la clase minoritaria más lejanos sea menor.
NearMiss 3: Seleccionamos de forma aleatoria una cantidad de elementos mayoritarios para cada elemento minoritario.

Neighbourhood Cleaning Rule.

"Improving Identification of Difficult Small Classes by Balancing Class Distribution" escrito por *J. Laurikkala* [6]

Eliminar vecinos que clasifican de forma errónea otras instancias.
ENN $\rightarrow$ Predecir clase con los tres vecinos más cercanos $\rightarrow$ Incorrecta: eliminamos sus vecinos.

One-Side Selection.

"Addressing the Curse of Imbalanced Training Sets: One-Side Selection" escrito por *Miroslav Kubat y Stan Matwin* [4]

Detectar ejemplos mal clasificados usando las instancias minoritarias y una mayoritaria al azar $\rightarrow$ Tomek Link: instancias minoritarias y las instancias mal clasificadas.

Random Undersampling.

Eliminar instancias mayoritarias de forma aleatoria.

Tomek Link.

"Two Modifications of CNN" escrito por *Ivan Tomek* [10]

Instancias ruidosas: vecino más cercano de la clase opuesta y si son el vecino más cercano entre sí, ambos elementos forman un *tomek-link*

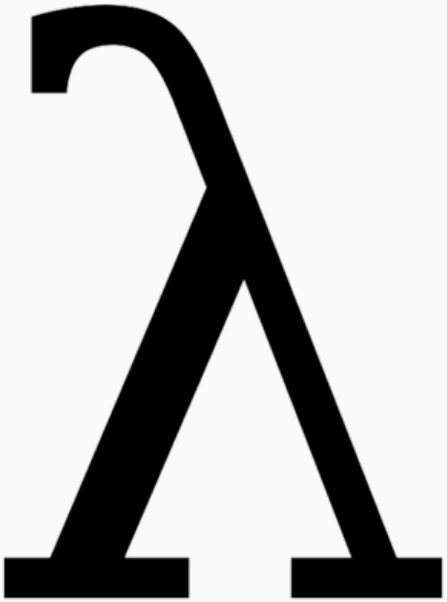
Undersampling Based on Clustering.

"Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset" escrito por *Show-Jane Yen y Yue-Shi Lee* [11]

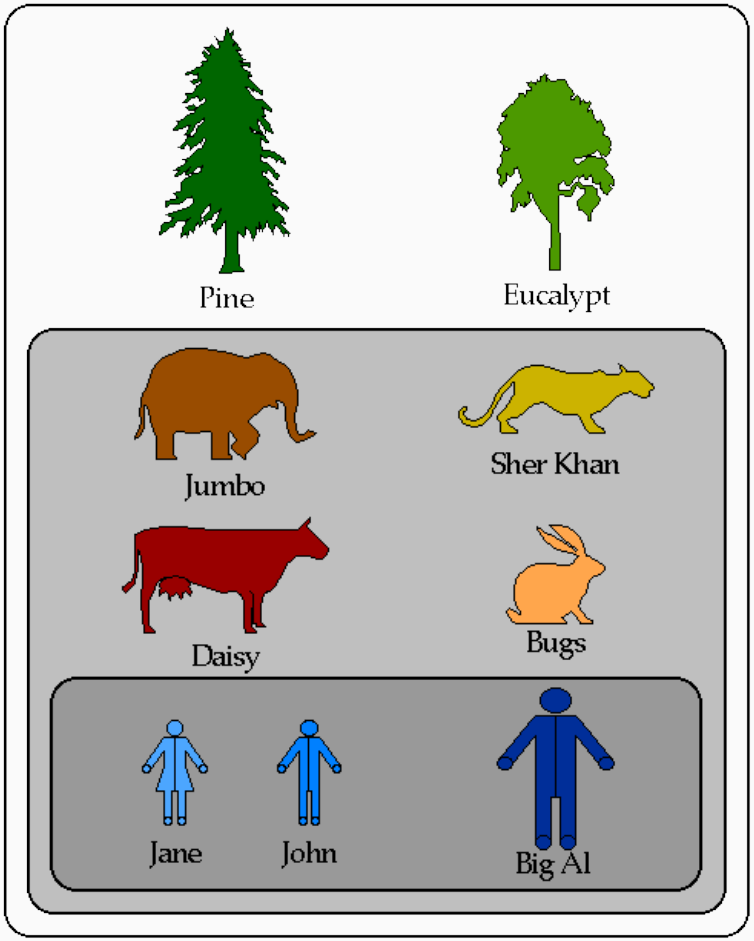
Conjunto de datos balanceado en cada *clúster* y devolver la unión de todos los subconjuntos balanceados formado en cada *clúster*

Scala.

Patrón de diseño mixto.



Functional Programming



Simplicidad.

```
val r = new scala.util.Random
val d = (0 until 10).map(_ => (0 until 10).map(_ => r.nextDouble))
```

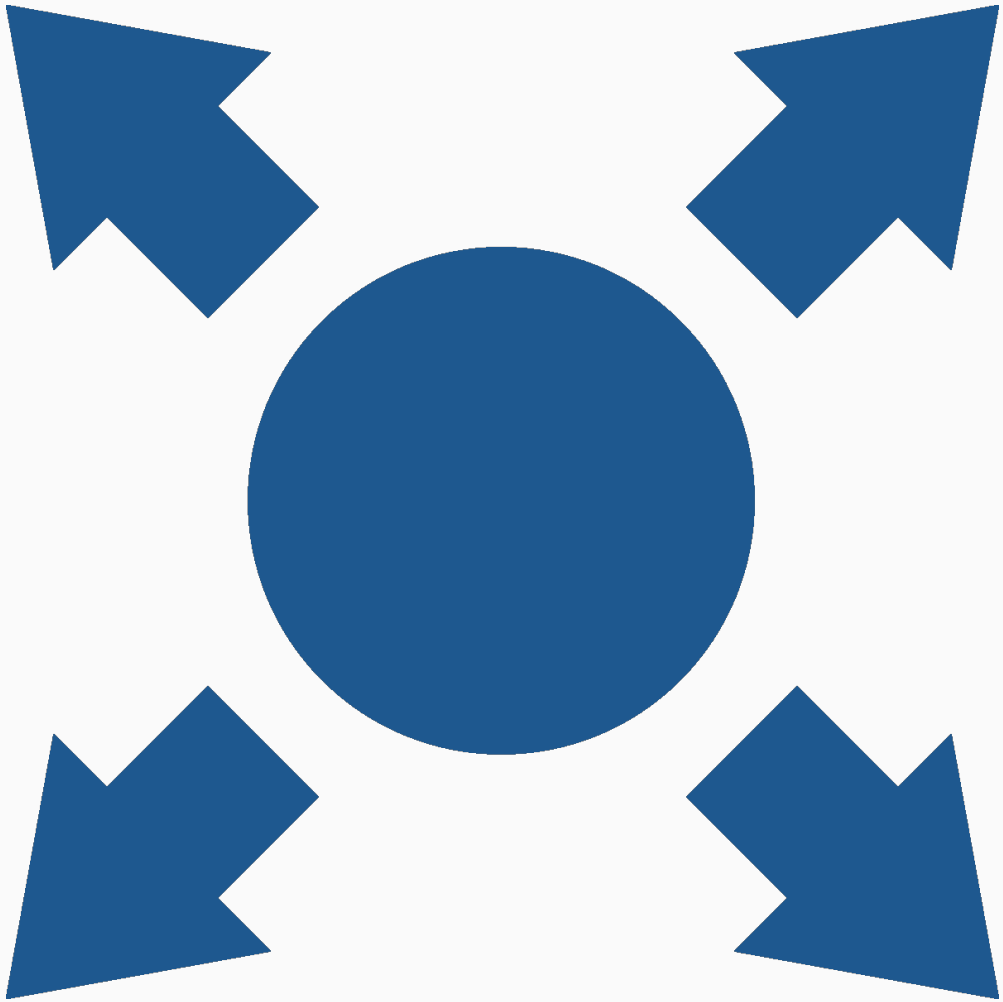
Simplicidad.

```
public class Persona {
    private final String nombre;
    private final String apellido;
    public Person(String nombre, String apellido) {
        this.nombre = nombre;
        this.apellido = apellido;
    }
    public String getNombre() {
        return nombre;
    }
    public String getApellido() {
        return apellido;
    }
}
```

Simplicidad.

```
class Persona(val nombre: String, val apellido: String)
```

Escalabilidad.



Modificadores de visibilidad.

- Publico.
- Protegido: *protected*.
- Privado: *private*.
- Protegido[scope]: *protected[scope]*.
- Privado[scope]: *private[scope]*.

Paralelismo.

```
scala> Array(0, 1, 2, 3, 4, 5).foreach(print)
res0: 012345
```

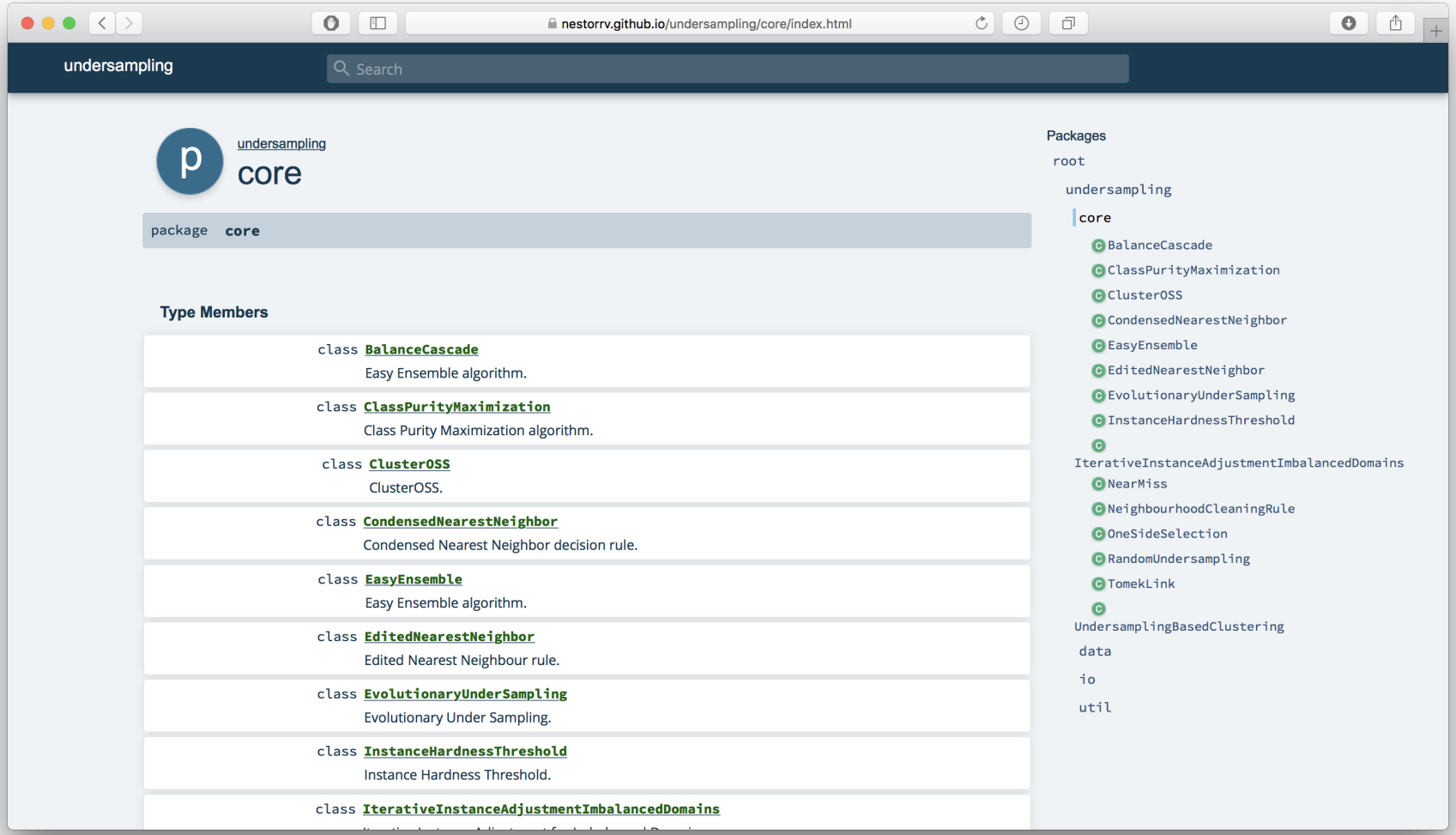
```
scala> Array(0, 1, 2, 3, 4, 5).par.foreach(print)
res1: 035421
```

Ausencia de Scala.

Descripción técnica.

Documentación.

https://nestorrv.github.io



sample.

```
sample(file: Option[String], ...)
```

Manual de usuario.

```
import undersampling.io.Reader
import undersampling.core.NeighbourhoodCleaningRule
import undersampling.io.Writer

val reader = new Reader
val writer = new Writer

val csvData = reader.readDelimitedText(file = pathToFile)
val arffData = reader.readArff(file = pathToFile)

val nclCSV = new NeighbourhoodCleaningRule(csvData, seed = 0L)
val resultCSV = nclCSV.sample(file = Option("milogCSV.log"))

val nclARFF = new NeighbourhoodCleaningRule(arffData, seed = 0L)
val resultARFF = nclARFF.sample(file = Option("milogARFF.log"))

writer.writeDelimitedText(file = "salidaCSV.csv", data = resultCSV)
writer.writeArff(file = "salidaARFF.arff", data = resultARFF)
```

Experimentación.

Porcentaje de reducción.

Dataset	BC	ClusterOSS	CNN	CPM	EE	ENN	EUS	IHTS	IPADE	NCL	NM	OSS	RU	SBC	TL	Media
ecoli-0_vs_1	30	70.91	92.27	91.82	30	0	30	51.82	91.36	4.55	30	42.27	30	4.09	4.55	40.24
ecoli1	54.17	83.33	76.19	74.7	54.17	4.46	54.17	73.81	88.99	11.01	54.17	32.74	54.17	13.39	11.9	49.42
ecoli2	69.05	89.88	80.06	81.25	69.05	2.08	69.05	81.25	92.86	5.65	69.05	16.37	69.05	8.04	7.14	53.99
ecoli3	79.17	87.5	80.95	85.12	79.17	3.57	79.17	86.01	90.18	6.55	79.17	25.6	79.17	9.82	7.74	58.59
glass-0-1-2-3_vs_4-5-6	52.34	85.98	85.51	86.92	52.34	3.74	52.34	73.83	85.51	8.41	52.34	8.41	52.34	5.61	5.61	47.41
glass0	34.58	80.84	63.55	66.36	34.58	14.49	34.58	54.21	82.24	20.09	34.58	37.38	34.58	17.29	20.56	41.99
glass1	28.97	84.11	59.35	62.15	28.97	10.28	28.97	46.26	81.31	20.09	28.97	23.83	28.97	13.55	19.63	37.69
glass6	72.9	83.18	91.12	89.72	72.9	0.47	72.9	84.11	87.85	7.48	72.9	59.35	72.9	5.61	5.61	58.6
haberman	47.06	82.03	46.73	51.96	47.06	10.78	46.73	73.2	87.58	23.2	47.06	34.31	47.06	31.7	30.72	47.15
iris0	33.33	91.33	98.67	98.67	33.33	0	33.33	54.67	91.33	0	33.33	67.33	33.33	1.33	2.67	44.84
new-thyroid1	67.44	80.93	94.88	93.95	67.44	0.93	67.44	68.37	88.84	6.05	67.44	71.63	67.44	3.26	4.65	56.71
new-thyroid2	67.44	83.26	93.02	92.56	67.44	0.93	67.44	81.4	89.3	5.58	67.44	56.74	67.44	4.19	4.65	56.59
page-blocks0	79.57	90.83	90.61	90.61	79.57	1.13	79.59	87.85	97.53	3.97	79.57	15.97	79.57	39.47	5.23	61.4
pima	30.21	79.04	50.91	51.3	30.21	11.72	30.34	62.63	92.84	24.48	30.21	29.95	30.21	30.21	27.34	40.77
segment0	71.49	87.56	97.01	97.18	71.49	0.78	71.75	68.59	98.35	1.34	71.49	20.23	71.49	20.71	5.89	57.02
vehicle0	52.96	82.51	84.28	81.8	52.96	2.01	52.84	59.22	91.37	6.97	52.96	14.42	52.96	18.44	13.24	47.93
vehicle1	48.7	77.9	55.08	54.37	48.7	10.64	50.12	65.37	80.97	21.87	48.7	27.66	48.7	33.22	26.95	46.6
vehicle2	48.46	80.97	83.69	83.1	48.46	3.55	48.82	58.27	93.74	7.45	48.46	13.83	48.46	27.19	13	47.16
vehicle3	49.88	84.99	57.92	56.26	49.88	8.51	50.71	45.86	84.52	20.33	49.88	27.9	49.88	34.04	26.24	46.45
wisconsin	30.01	95.9	89.17	89.31	30.01	1.61	30.01	38.36	92.39	3.07	30.01	55.64	30.01	0.29	2.93	41.25
yeast1	42.18	80.19	54.11	53.1	42.18	10.71	42.12	65.77	89.89	22.78	42.18	25.13	42.18	0.2	24.93	42.51
yeast3	78.03	89.82	83.49	83.96	78.03	2.22	78.03	87.94	93.8	6	78.03	8.22	78.03	25.27	8.22	58.61

Imbalanced Ratio.

Dataset	IR Original	BC	ClusterOSS	CNN	CPM	EE	ENN	EUS	IHTS	IPADE	NCL	NM	OSS	RU	SBC	TL	Media
ecoli-0_vs_1	1.85714	1	0.30612	1.125	2	1	1.85714	1	0.37662	0.72727	1.72727	1	0.76389	1	1.74026	1.91667	1.16935
ecoli1	3.36364	1	1.66667	1.10526	1.02381	1	3.16883	1	0.14286	0.54167	2.98667	1	2.42424	1	2.77922	3.16901	1.60055
ecoli2	5.46154	1	3.25	2.35	2.70588	1	5.32692	1	0.21154	0.5	5.21569	1	4.73469	1	4.94231	5.93333	2.67802
ecoli3	8.6	1	0.5	1.56	1.27273	1	8.25714	1	0.34286	0.73684	8.23529	1	6.57576	1	7.65714	9.33333	3.29807
glass-0-1-2-3_vs_4-5-6	3.19608	1	2	0.9375	1.15385	1	3.03922	1	0.09804	0.47619	2.92	1	3.17021	1	2.96078	3.3913	1.67647
glass0	2.05714	1	1.92857	1.51613	1.48276	1	1.61429	1	0.4	1.11111	1.89831	1	1.57692	1	1.52857	1.83333	1.326
glass1	1.81579	1	0.61905	1.175	1.25	1	1.52632	1	0.51316	0.66667	1.375	1	1.76271	1	1.43421	1.77419	1.13975
glass6	6.37931	1	0.56522	2.16667	3.4	1	6.34483	1	0.17241	0.85714	5.82759	1	2.22222	1	5.96552	6.21429	2.58239
haberman	2.77778	1	0.89655	1.50769	1.29688	1	2.37037	1.01235	0.01235	0.52	2.26389	1	2.65455	1	1.58025	2.65517	1.38467
iris0	2	1	12	1	1	1	2	1	0.36	0.18182	2	1	0.02083	1	1.96	2.04167	1.83762
new-thyroid1	5.14286	1	0.51852	1.2	0.85714	1	5.08571	1	0.94286	0.33333	4.94118	1	1.10345	1	4.94286	5.21212	2.00914
new-thyroid2	5.14286	1	0.44	1.14286	0.77778	1	5.08571	1	0.14286	0.35294	4.97059	1	1.90625	1	4.88571	5.21212	1.99445
page-blocks0	8.78891	1	0.0308	1.41315	1.45933	1	8.678	0.99821	0.18962	0.90141	8.51993	1	7.65913	1	4.92487	8.95393	3.18189
pima	1.86567	1	0.14184	1.17919	1.14943	1	1.52985	0.99627	0.0709	0.77419	1.46809	1	1.69	1	1	1.87629	1.0584
segment0	6.0152	1	0.05515	2.13636	2.42105	1	5.96049	0.98176	1.20365	0.72727	6.00615	1	5.05592	1	4.56231	5.96154	2.60478
vehicle0	3.25126	1	0.0963	1.33333	1.02632	1	3.16583	1.00503	0.73367	0.78049	3.12042	1	3.02222	1	2.46734	3.44848	1.6133
vehicle1	2.89862	1	1.28049	1.55034	1.57333	1	2.48387	0.9447	0.35023	1.26761	2.35533	1	2.77778	1	1.60369	3.03922	1.54844
vehicle2	2.88073	1	0.11034	1.875	1.34426	1	2.74312	0.98624	0.61927	0.60606	2.72857	1	2.83684	1	1.82569	2.8534	1.50192
vehicle3	2.99057	1	0.64935	1.45517	1.46667	1	2.65094	0.96698	1.16038	1.01538	2.51042	1	2.74233	1	1.63208	2.78182	1.53543
wisconsin	1.85774	1	0.64706	0.39623	0.40385	1	1.81172	1	0.76151	0.625	1.79325	1	0.35874	1	1.84937	1.84549	1.03281
yeast1	2.45921	1	1.9697	1.42349	1.31229	1	2.08858	1.00233	0.18415	0.78571	2.07239	1	2.43963	1	2.45221	2.49216	1.48151
yeast3	8.10429	1	0.10219	2.02469	1.8	1	7.90184	1	0.09816	1.2439	7.77358	1	8.2027	1	5.80368	8.01987	3.19804

k-fold.

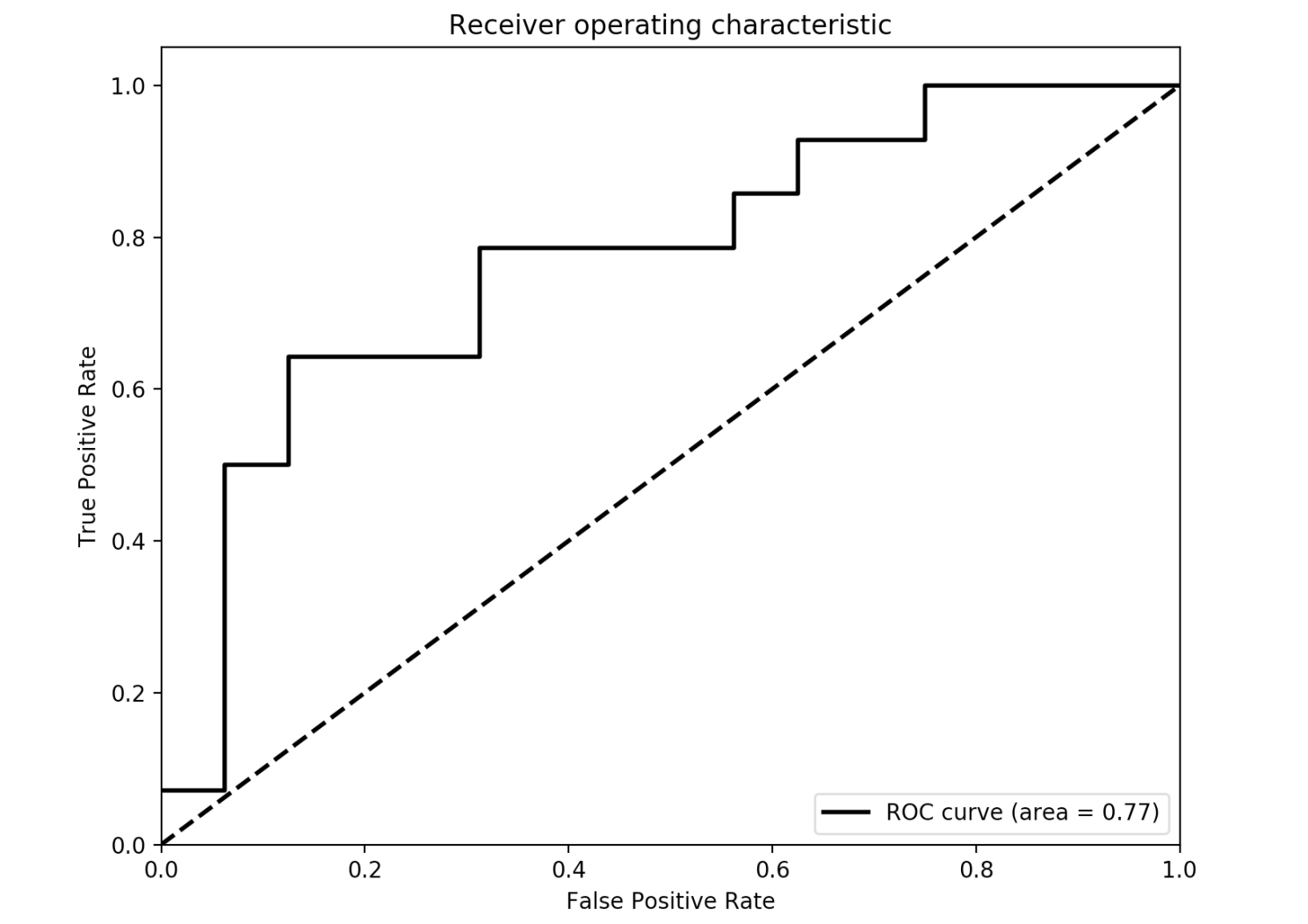
Iteracion 1	Test	Train	Train	Train	Train
Iteracion 2	Train	Test	Train	Train	Train
Iteracion 3	Train	Train	Test	Train	Train
Iteracion 4	Train	Train	Train	Test	Train
Iteracion 5	Train	Train	Train	Train	Test

Matriz de confusión.

		Predicción	
		Positivo	**Negativo**
Real	**Positivo**	TP	FN
	Negativo	FP	TN

AUC - ROC: Receiver Operating Characteristic.

$$TPR = TP/(TP + FN) \text{ — } FPR = FN/(FP + TN)$$



Media geométrica.

$$g = \sqrt{a^+ \cdot a^-}$$

donde a^+ denota la precisión en las instancias positivas —esto es *TPR*— y a^- denota la precisión en las instancias positivas —esto es *TNR*—, y se calcula de la siguiente forma:

$$TNR = \frac{TP}{TP + FN} \tag{1}$$

Porcentaje de acierto.

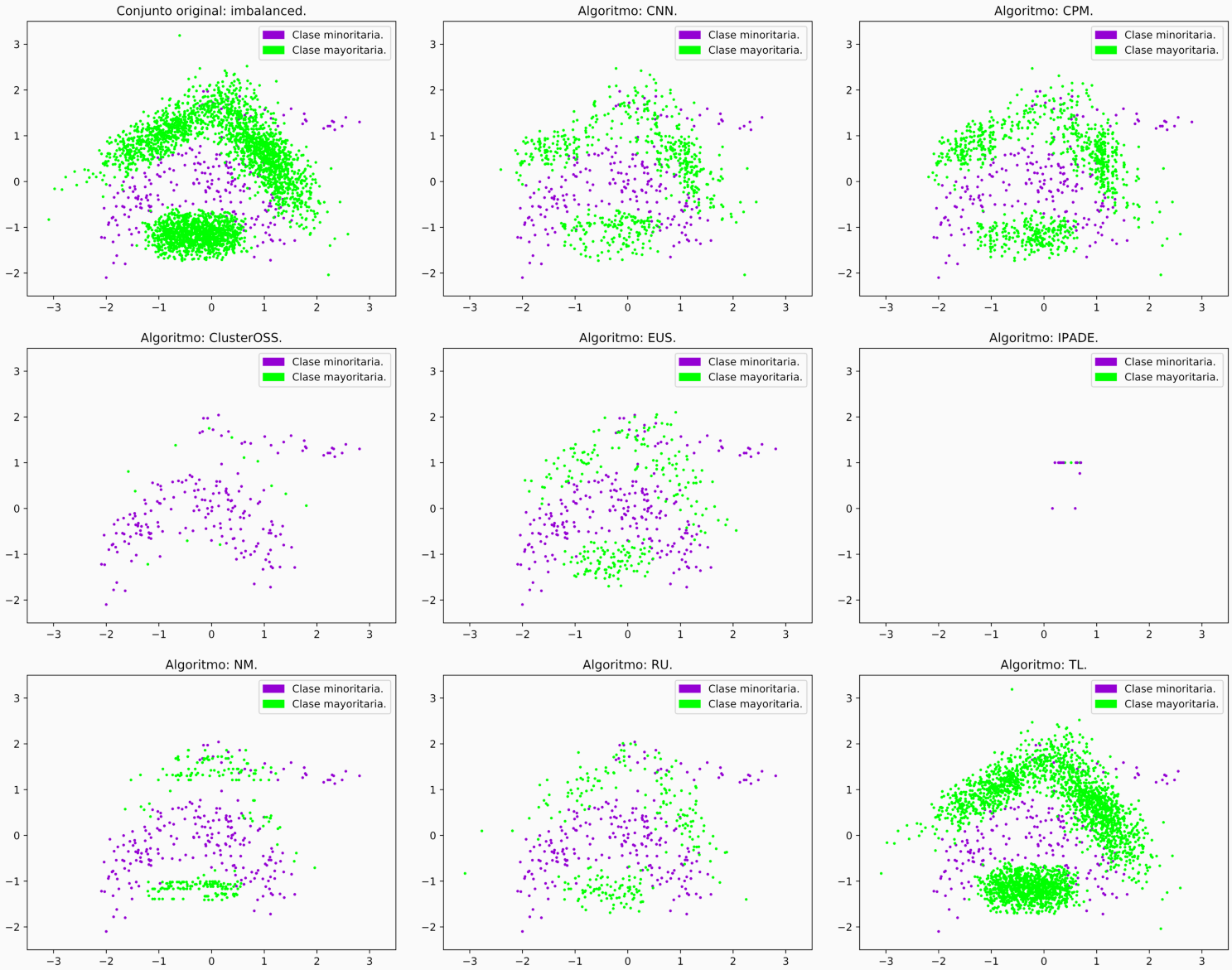
1	0	1	1	0
1	1	0	1	0
✓	✗	✗	✓	✓

60 %

ecoli3.

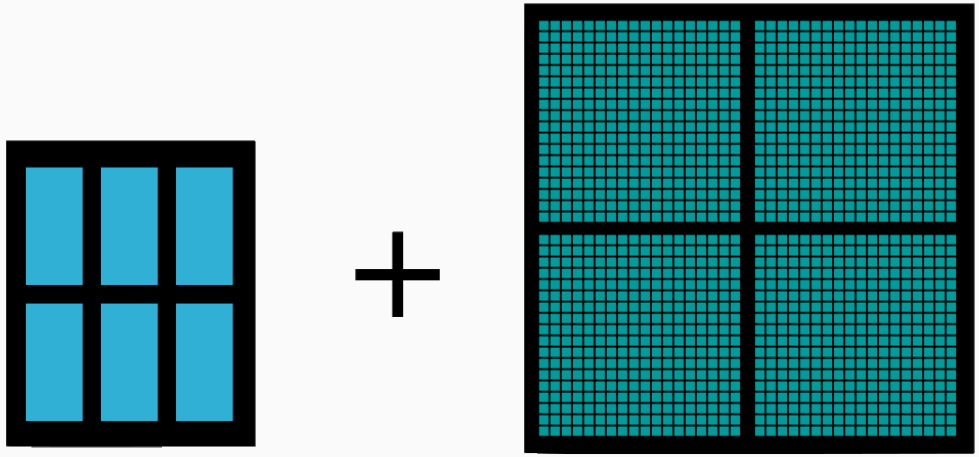
Algoritmo	j48auc	j48gm	j48_%	SVMauc	SVMgm	SVM_%
BC	0.768	0.751	79.485	0.87	0.866	81.287
CNN	0.653	0.512	91.085	0.5	0	89.577
CPM	0.657	0.522	64.007	0.557	0.274	88.401
ClusterOSS	0.807	0.757	74.522	0.782	0.678	81.232
EE	0.875	0.858	81.985	0.869	0.866	81.121
ENN	0.823	0.765	92.5	0.661	0.565	91.36
EUS	0.84	0.836	83.272	0.884	0.879	81.544
IHTS	0.591	0.408	49.118	0.697	0.495	54.706
IPADE-ID	0.908	0.857	87.61	0.715	0.581	71.581
NCL	0.803	0.715	92.555	0.514	0.076	89.871
NM	0.706	0.61	74.853	0.643	0.636	58.658
OSS	0.8	0.707	91.415	0.509	0.074	88.989
RU	0.823	0.811	80.68	0.863	0.858	80.074
SCB	0.862	0.778	89.908	0.818	0.805	92.279
TL	0.82	0.769	91.985	0.5	0	89.577
original	**0.805**	**0.753**	**93.438**	**0.5**	**0**	**89.577**

Nube de puntos.



Trabajos futuros.

Trabajos futuros.



CPU
Multiple Cores

GPU
Thousands of Cores

References. i

V. H. Barella, E. P. Costa, and A. C. P. L. F. Carvalho.

Clusteross: a new undersampling method for imbalanced learning.

2014.

S. García and F. Herrera.

Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy.

Evol. Comput., 17(3):275–306, Sept. 2009.

P. Hart.

The condensed nearest neighbor rule (corresp.).

IEEE Trans. Inf. Theor., 14(3):515–516, Sept. 2006.

M. Kubat and S. Matwin.

Addressing the curse of imbalanced training sets: One-sided selection.

In *In Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

D. L. Wilson.

Asymptotic properties of nearest neighbor rules using edited data.

2:408 – 421, 08 1972.

J. Laurikkala.

Improving identification of difficult small classes by balancing class distribution.

In *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*, AIME '01, pages 63–66, London, UK, UK, 2001. Springer-Verlag.

X.-Y. Liu, J. Wu, and Z.-H. Zhou.

Exploratory undersampling for class-imbalance learning.

Trans. Sys. Man Cyber. Part B, 39(2):539–550, Apr. 2009.

References. ii

V. López, I. Triguero, C. J. Carmona, S. García, and F. Herrera.

Addressing imbalanced classification with instance generation techniques: Ipade-id.

Neurocomput., 126:15–28, Feb. 2014.

M. R. Smith, T. Martinez, and C. G. Giraud-Carrier.

An empirical study of instance hardness.

2011.

I. Tomek.

Two modifications of cnn.

6, 11 1976.

S.-J. Yen and Y.-S. Lee.

Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset, pages 731–740.

Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

K. Yoon and S. Kwek.

An unsupervised learning approach to resolving the data imbalanced issue in supervised learning problems in functional genomics.

In *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems*, HIS '05, pages 303–308, Washington, DC, USA, 2005. IEEE Computer Society.

J. Zhang and I. Mani.

KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction.

In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.

Gracias.