# **Analítica de Datos en Big Data**

# Primeros experimentos

Los experimentos realizados para comenzar han tenido los siguientes datasets:

* 1. Datos originales con 499449 elementos en train y 412636 en test, y 631 características, de donde:
  + Train: Clase 0 -> 349272 y clase 1 -> 150177
  + Test: Clase 0 -> 275091 y clase 1 -> 137545
* 2. Igual que el original, pero con los datos discretizados y con selección de 10 características.

Además, los modelos usados son:

* RF: Random Forest con 10 árboles, máxima profundidad de 4 y un max Bins de 32.
* NB: Naive Bayes con un lambda igual a 1.0
* DT: Decision Trees con máxima profundidad de 4 y un max Bins de 32.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos + Modelo | Resultados | | | | | | | | Tiempo (s) |
| Test Error % | Precisión % | ROC % | TP | FP | FN | TN | TPR x TNR |
| 1 RF | 29,363 | 70,636 | 57,447 | 266876 | 112951 | 8215 | 24594 | 0,1735 | 19 |
| 1 NB | 29,374 | 70,625 | 60,726 | 248743 | 94863 | 26348 | 42682 | 0,2806 | 4 |
| 1 DT | 27,044 | 72,955 | 67,15 | 232632 | 69137 | 42459 | 68408 | 0,4206 | 15 |
| 2 RF | 30,027 | 69,972 | 57,079 | 263420 | 112235 | 11671 | 25310 | 0,1762 | 7 |
| 2 NB | 29,629 | 70,37 | 65,136 | 222376 | 69548 | 52715 | 67997 | 0,3996 | 2 |
| 2 DT | 29,104 | 70,895 | 61,767 | 245247 | 90251 | 29844 | 47294 | 0,3065 | 3 |

Donde podemos ver como los resultados, han mejorado en algunos casos y han empeorado en otros, por lo que vamos a seguir preprocesando. Además, hemos guardado en nuestro directorio el dataset reducido con 10 características para volver a cargarlo y ver que se obtienen los mismos resultados, de forma que ha sido así.

Tenemos en los datos un imbalanceo, ya que tenemos 70-30 de ratio. Probaremos a hacer undersampling y oversampling para ver qué tal se comportan los modelos.

# Aplicación de RUS y ROS

En este apartado vamos a realizar balanceo de los datos con RUS, es decir, vamos a eliminar instancias de la clase mayoritaria para balancear el dataset. Además, vamos a aplicar ROS, es decir, vamos a crear instancias de la clase minoritaria para no tener desbalanceo de los datos. ROS lo he aplicado para tener el mismo número de instancias de las dos clases.

Por lo tanto, lo primero que se ha realizado ha sido el undersampling, quedando un dataset con los siguientes elementos en train:

* 149957 de la clase 0
* 150177 de la clase 1.

Por lo que hemos solucionado este problema de imbalanceo de los datos. Ahora, vamos a realizar las mismas pruebas que en el apartado anterior, de forma que vamos a tener los datasets:

* 3: Referente al dataset con undersampling y las 631 características.
* 4: Dataset con undersampling, pero haciendo selección de 10 características.

Los resultados obtenidos han sido:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos + Modelo | Resultados | | | | | | | | Tiempo (s) |
| Test Error % | Precisión % | ROC % | TP | FP | FN | TN | TPR x TNR |
| 3 RF | 33,254 | 66,745 | 68,567 | 173584 | 35715 | 101507 | 101830 | 0,4672 | 19 |
| 3 NB | 35,569 | 64,43 | 64,486 | 176936 | 48618 | 98155 | 88927 | 0,4158 | 5 |
| 3 DT | 31,384 | 68,615 | 69,958 | 181366 | 35780 | 93725 | 101765 | 0,4878 | 13 |
| 4 RF | 34,796 | 65,203 | 67,738 | 165420 | 33913 | 109671 | 103632 | 0,4531 | 7 |
| 4 NB | 33,172 | 66,827 | 67,384 | 180772 | 42562 | 94319 | 94983 | 0,4538 | 2 |
| 4 DT | 31,603 | 68,396 | 67,079 | 195393 | 50711 | 79698 | 86834 | 0,4484 | 4 |

A la vista de estos resultados, vemos como la precisión ha empeorado, pero la medida TPR x TNR ha aumentado bastante, y la medida ROC ha aumentado.

Ahora vamos a realizar oversampling, realizando los mismos experimentos. En train, una vez realizado ROS, tenemos:

* 349272 de la clase 0.
* 349742 de la clase 1.

Por lo que ya hemos solucionado el imbalanceo de los datos. Si ahora tenemos los datasets:

* 5: Referente al dataset con oversampling y las 631 características.
* 6: Dataset con oversampling, pero haciendo selección de 10 características.

Los resultados obtenidos han sido:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos + Modelo | Resultados | | | | | | | | Tiempo (s) |
| Test Error % | Precisión % | ROC % | TP | FP | FN | TN | TPR x TNR |
| 5 RF | 33,065 | 66,934 | 68,673 | 174565 | 35914 | 100526 | 101631 | 0,4689 | 25 |
| 5 NB | 35,519 | 64,48 | 64,515 | 177183 | 48660 | 97908 | 88885 | 0,4162 | 5 |
| 5 DT | 31,56 | 68,439 | 69,935 | 180044 | 35182 | 95047 | 102363 | 0,4871 | 19 |
| 6 RF | 32,48 | 67,519 | 68,289 | 181501 | 40437 | 93590 | 97108 | 0,4658 | 7 |
| 6 NB | 32,413 | 67,586 | 67,931 | 184029 | 42686 | 91062 | 94859 | 0,4614 | 2 |
| 6 DT | 31,729 | 68,27 | 68,007 | 189250 | 45087 | 85841 | 92458 | 0,4624 | 3 |

Donde podemos ver como los resultados, con respecto a RUS, son un poco mejores con selección de características. Vamos a seguir, por lo tanto, haciendo pruebas preprocesando el dataset.

# Selección de características 200

El siguiente experimento es sobre selección de características, 200 concretamente, por lo que tenemos el siguiente conjunto de datos:

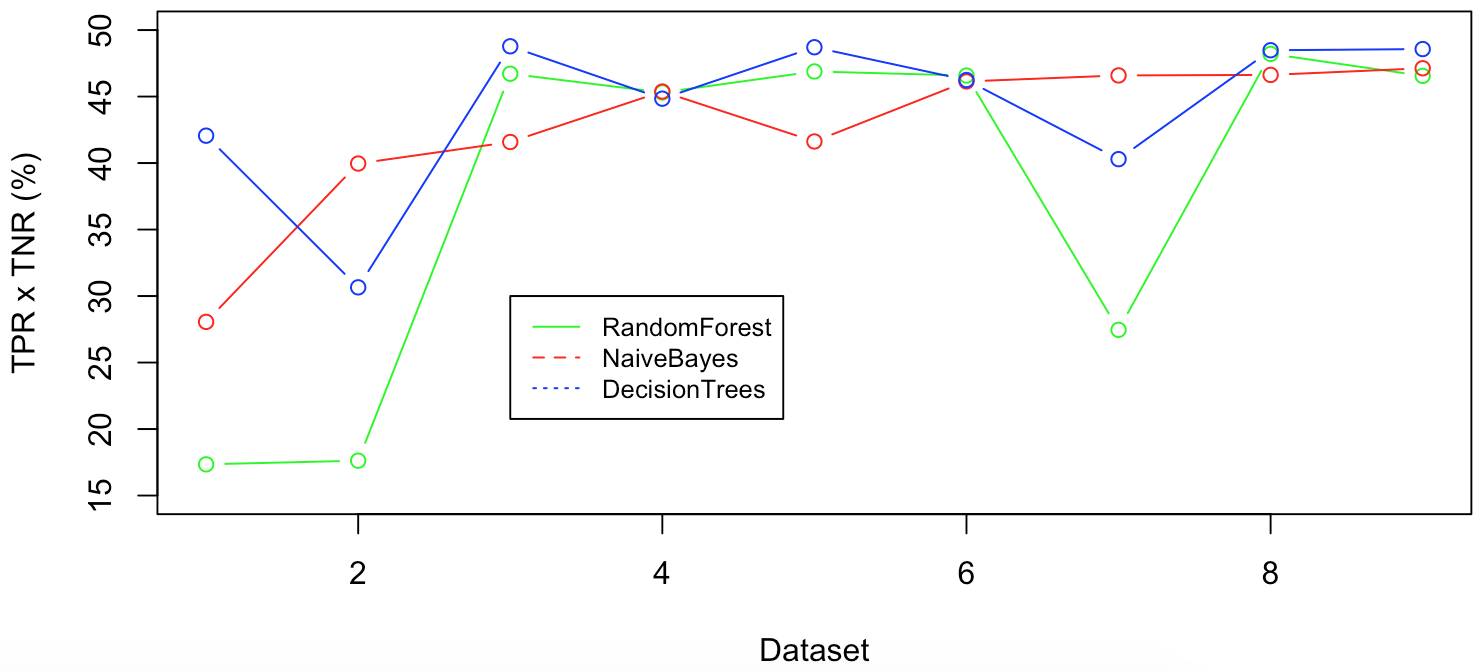
* 7: Referente al dataset original con una selección de 200 características.
* 8: Dataset con RUS y una selección de 200 características.
* 9: Dataset con ROS y una selección de 200 características.

Por lo que los resultados obtenidos han sido:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos + Modelo | Resultados | | | | | | | | Tiempo (s) |
| Test Error % | Precisión % | ROC % | TP | FP | FN | TN | TPR x TNR |
| 7 RF | 27,688 | 72,311 | 61,548 | 258134 | 97297 | 16957 | 40248 | 0,2746 | 9 |
| 7 NB | 28,77 | 71,229 | 68,681 | 209962 | 53589 | 65129 | 83956 | 0,4659 | 3 |
| 7 DT | 27,207 | 72,792 | 66,351 | 235677 | 72856 | 39414 | 64689 | 0,4029 | 5 |
| 8 RF | 31,496 | 68,503 | 69,495 | 182995 | 37868 | 92096 | 99677 | 0,4821 | 9 |
| 8 NB | 31,368 | 68,631 | 68,292 | 190670 | 45015 | 84421 | 92530 | 0,4663 | 4 |
| 8 DT | 31,913 | 68,086 | 69,823 | 177744 | 34339 | 97347 | 103206 | 0,4848 | 5 |
| 9 RF | 32,908 | 67,091 | 68,333 | 177729 | 38430 | 97362 | 99115 | 0,4656 | 11 |
| 9 NB | 30,819 | 69,18 | 68,666 | 193134 | 45217 | 81957 | 92328 | 0,4713 | 4 |
| 9 DT | 31,788 | 68,211 | 69,87 | 178517 | 34597 | 96574 | 102948 | 0,4857 | 7 |

Si nos enfocamos en la medida TPR x TNR, vemos como con el conjunto original, se han obtenido resultados peores que con el uso de ROS y RUS. Con respecto a ROS y RUS, de momento se han obtenido resultados semejantes a los vistos anteriormente.

Veamos ahora una gráfica sobre cómo van los experimentos en función de la medida TPR x TNR:



De forma que vemos como el modelo que mejores resultados ha dado hasta ahora es el Decision Trees en relación a la métrica TPR x TNR. Los mejores resultados se obtienen en los Datasets 3, 5, 8 y 9, que corresponden a los datasets originales y con selección de 200 características usando ROS y RUS. Podemos decir, además, que la selección de las 200 características ha dado un resultado muy parecido al completo. Viendo esto, voy a realizar más selección de características para intentar conseguir mejores resultados o parecidos.

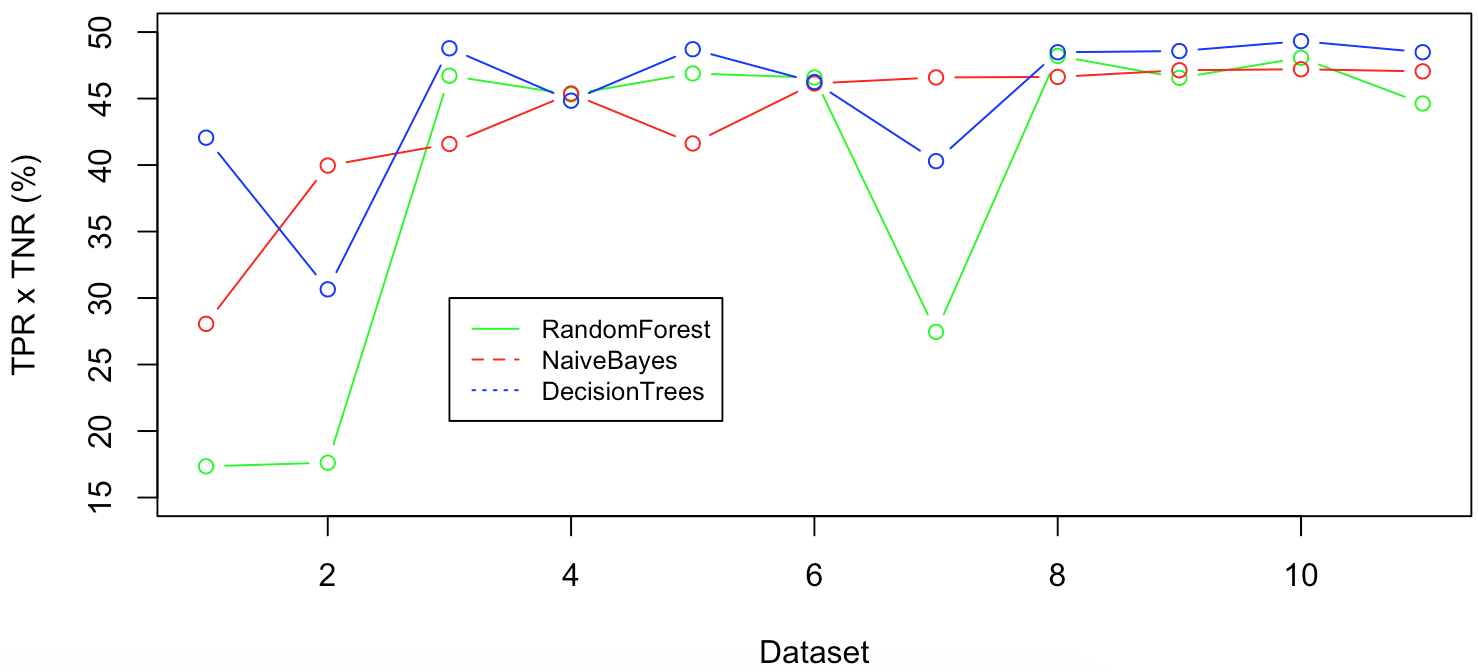
# Selección de características 150

En este apartado voy a probar la selección de 150 características en los dataset habiendo realizado ROS y RUS para equilibrar los datos al mismo nivel en cada clase. Por lo tanto, vamos a tener los datasets:

* 10: Dataset con selección de 150 características con ROS
* 11: Dataset con selección de 150 características con RUS

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos + Modelo | Resultados | | | | | | | | Tiempo (s) |
| Test Error % | Precisión % | ROC % | TP | FP | FN | TN | TPR x TNR |
| 10 RF | 29,156 | 70,843 | 69,453 | 202531 | 47751 | 72560 | 89794 | 0,4806 | 9 |
| 10 NB | 30,867 | 69,132 | 68,716 | 192471 | 44749 | 82620 | 92796 | 0,4720 | 3 |
| 10 DT | 30,229 | 69,77 | 70,24 | 189343 | 38991 | 85748 | 98554 | 0,4932 | 6 |
| 11 RF | 35,562 | 64,437 | 67,388 | 161024 | 32678 | 114067 | 104867 | 0,4463 | 8 |
| 11 NB | 30,979 | 69,02 | 68,6 | 192178 | 44920 | 82913 | 92625 | 0,4704 | 3 |
| 11 DT | 31,913 | 68,086 | 69,823 | 177744 | 34339 | 97347 | 103206 | 0,4848 | 5 |

Veamos la gráfica para comparar los resultados de una forma gráfica:



Por lo tanto, vemos como el Dataset con 150 características ha funcionado un poco mejor que con 200 características.

El siguiente paso a probar será el uso de ROS con diferentes tasas de balanceo. Anteriormente la tasa que se había usado era 100, de forma que se tenían el mismo número de valores para la clase minoritaria que la mayoritaria. Si usamos 90, tendremos unos pocos valores en la minoritaria menos que en la mayoritaria. Además, usaré 110, 120 y 130.

# Aplicación de ROS

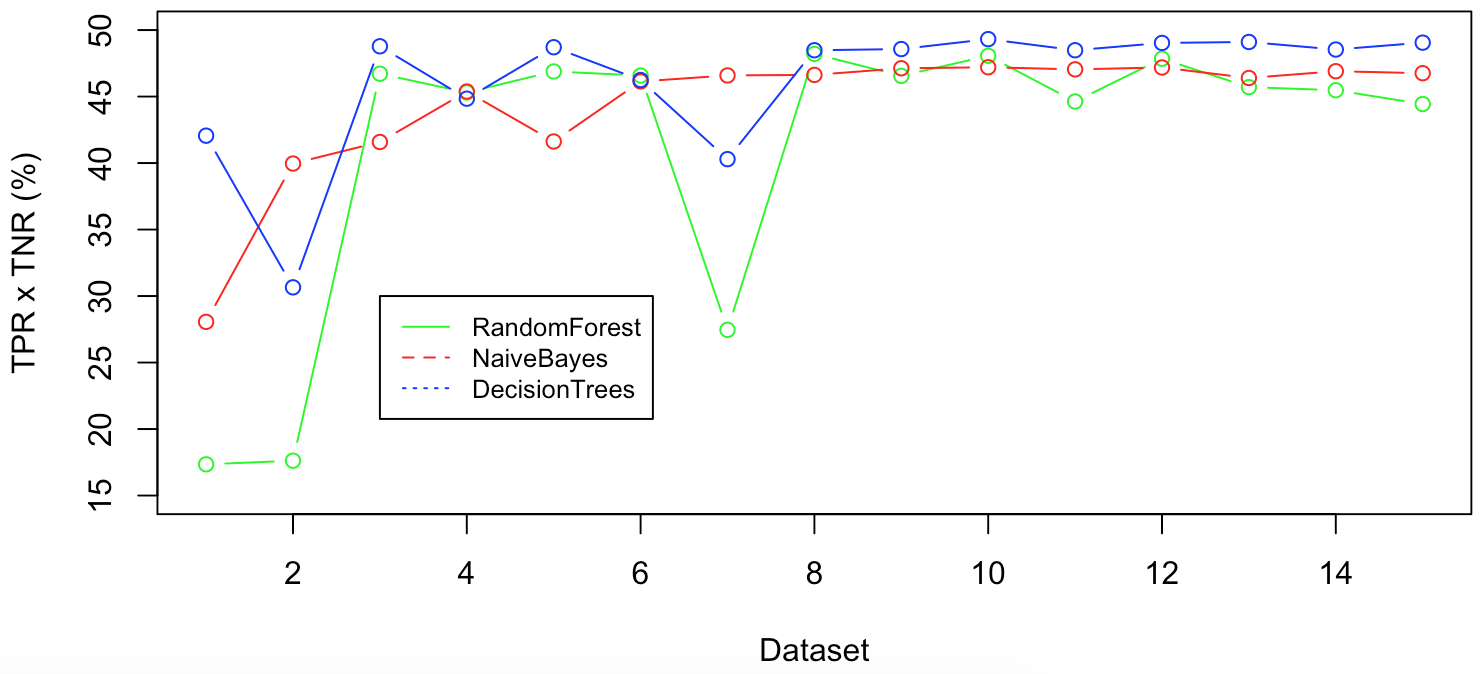
Como he comentado, voy a usar ROS con valores de 90, 110, 120 y 130. Además, todos ellos serán sobre el dataset con 150 características para acelerar los cálculos. Estos dataset son:

* 12: ROS 90, quedando 349272 de la clase 0 y 314419 de la clase 1 en train.
* 13: ROS 110, quedando 349272 de la clase 0 y 384382 de la clase 1 en train.
* 14: ROS 120, quedando 349272 de la clase 0 y 419850 de la clase 1 en train.
* 15: ROS 130, quedando 349272 de la clase 0 y 454160 de la clase 1 en train.

Veamos los resultados:

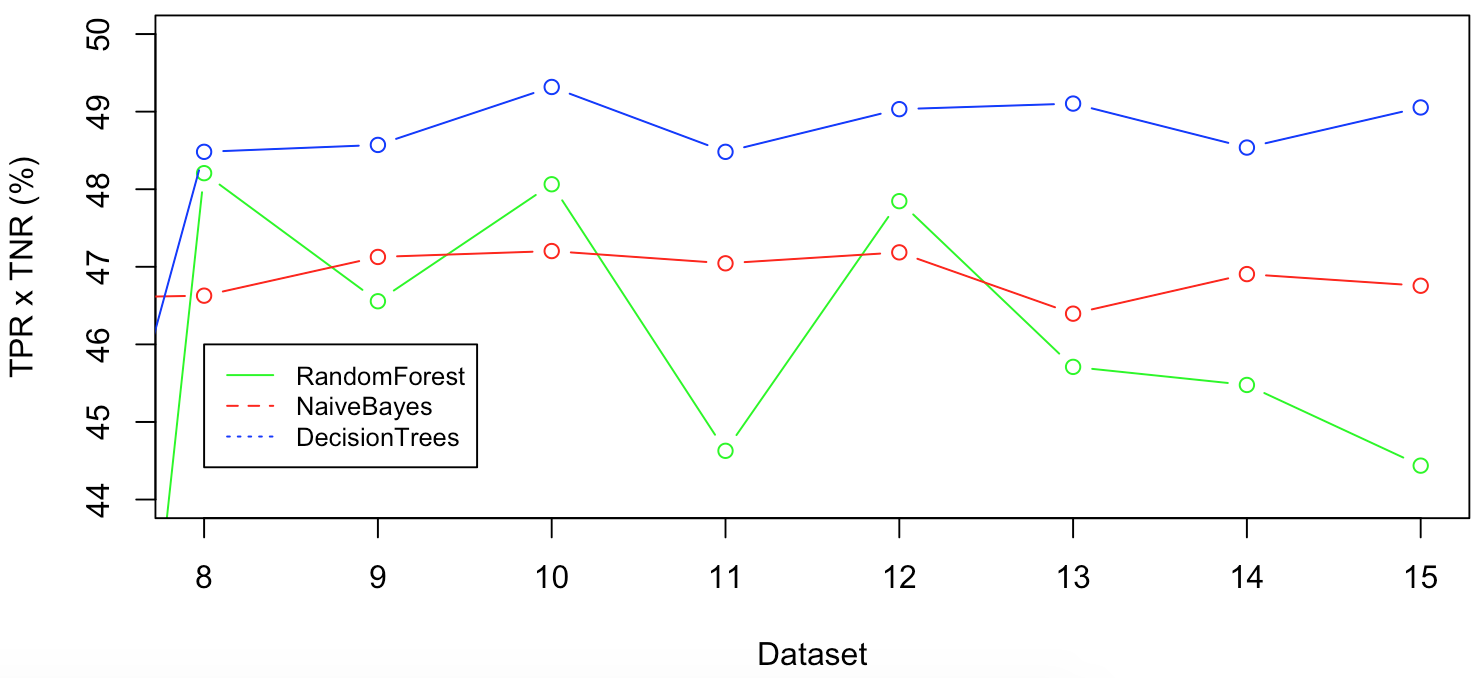
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos + Modelo | Resultados | | | | | | | | Tiempo (s) |
| Test Error % | Precisión % | ROC % | TP | FP | FN | TN | TPR x TNR |
| 12 RF | 29,504 | 70,495 | 69,268 | 200680 | 47334 | 74411 | 90211 | 0,4785 | 9 |
| 12 NB | 30,491 | 69,508 | 68,731 | 195487 | 46214 | 79604 | 91331 | 0,4719 | 3 |
| 12 DT | 30,978 | 69,021 | 70,096 | 183958 | 36694 | 91133 | 100851 | 0,4903 | 6 |
| 13 RF | 34,919 | 65,08 | 68,291 | 161367 | 30365 | 113724 | 107180 | 0,4571 | 9 |
| 13 NB | 31,615 | 68,384 | 68,119 | 189584 | 44948 | 85507 | 92597 | 0,4640 | 3 |
| 13 DT | 30,892 | 69,107 | 70,143 | 184410 | 36793 | 90681 | 100752 | 0,4910 | 6 |
| 14 RF | 35,176 | 64,823 | 68,187 | 159815 | 29874 | 115276 | 107671 | 0,4548 | 10 |
| 14 RF | 31,652 | 68,347 | 68,489 | 187237 | 42756 | 87854 | 94789 | 0,4691 | 3 |
| 14 NB | 31,784 | 68,215 | 69,838 | 178725 | 34788 | 96366 | 102757 | 0,4854 | 5 |
| 15 DT | 36,558 | 63,441 | 68,126 | 148749 | 24511 | 126342 | 113034 | 0,4444 | 10 |
| 15 NB | 31,696 | 68,303 | 68,378 | 187485 | 43185 | 87606 | 94360 | 0,4676 | 3 |
| 15 DT | 31,024 | 68,975 | 70,123 | 183426 | 36355 | 91665 | 101190 | 0,4905 | 6 |

Si observamos la siguiente gráfica sobre TPR x TNR en todos los experimentos realizados, tenemos:



Por lo que estos últimos experimentos con ROS han mejorado los resultados.

Podemos observar que los mejores resultados se obtienen a partir del Dataset 8, por lo que vamos a visualizar la gráfica a partir de ese dataset:



Repasemos que dataset son estos:

* 8: Dataset con RUS y una selección de 200 características.
* 9: Dataset con ROS y una selección de 200 características.
* 10: Dataset con selección de 150 características con ROS
* 11: Dataset con selección de 150 características con RUS
* 12: ROS 90, quedando 349272 de la clase 0 y 314419 de la clase 1 en train.
* 13: ROS 110, quedando 349272 de la clase 0 y 384382 de la clase 1 en train.
* 14: ROS 120, quedando 349272 de la clase 0 y 419850 de la clase 1 en train.
* 15: ROS 130, quedando 349272 de la clase 0 y 454160 de la clase 1 en train.

Es decir, todos ellos tienen que ver con la aplicación de ROS y RUS. Veamos, para cada uno de estos métodos cual podemos seleccionar.

Aparentemente, el mejor modelo de RUS es el dataset 8 ya que, en el 11, Random Forest cae bastante. Para ROS podemos seleccionar como el mejor el dataset 10, ya que por lo general produce mejores resultados para los 3 algoritmos, es decir el dataset con ROS a un nivel de 100, igualando los elementos de cada clase, y selección de características de 150.

En el siguiente paso, vamos a aumentar el número de árboles en los modelos y su profundidad, y además haremos una selección de características de 100 elementos para ver el comportamiento.

# Modelos

Ahora vamos a trabajar sobre los 2 dataset descritos anteriormente:

8: Dataset con RUS y una selección de 200 características.

10: Dataset con ROS, equilibrando al mismo número los elementos de cada clase, y una selección de 150 características.

Por lo tanto, vamos a dividir esta sección en dos apartados, uno para ROS y otro para RUS. Además, todas las pruebas las realizaré modificando los parámetros de Random Forest y Decisión Trees que serán los algoritmos con los que trabajaré. Estos parámetros serán de 50 árboles y máxima profundidad igual a 16 para los dos modelos.

## ROS

Las pruebas que voy a realizar, sobre el conjunto equilibrado con ROS, para tener los mismos elementos de cada clase, son:

* 1: Selección de 200 características
* 2: Selección de 150 características
* 3: Selección de 125 características
* 4: Selección de 100 características

Obteniendo los siguientes resultados:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos + Modelo | Resultados | | | | | | | | Tiempo (s) |
| Test Error % | Precisión % | ROC % | TP | FP | FN | TN | TPR x TNR |
| 1 RF | 25,324 | 74,675 | 73,542 | 211657 | 41066 | 63434 | 96479 | 0,5397 | 342 |
| 1 DT | 28,477 | 71,522 | 70,366 | 203112 | 45529 | 71979 | 92016 | 0,4939 | 20 |
| 2 RF | 26,03 | 73,969 | 73,036 | 208620 | 40939 | 66471 | 96606 | 0,5326 | 309 |
| 2 DT | 28,721 | 71,278 | 70,635 | 199620 | 43044 | 75471 | 94501 | 0,4986 | 17 |
| 3 RF | 26,354 | 73,645 | 72,866 | 206882 | 40538 | 68209 | 97007 | 0,5304 | 280 |
| 3 DT | 28,725 | 71,274 | 70,476 | 200462 | 43903 | 74629 | 93642 | 0,4961 | 14 |
| 4 RF | 26,893 | 73,106 | 72,36 | 205210 | 41093 | 69881 | 96452 | 0,5231 | 268 |
| 4 DT | 28,854 | 71,145 | 70,092 | 201509 | 45482 | 73582 | 92063 | 0,4903 | 13 |

## RUS

Las pruebas que voy a realizar, sobre el conjunto equilibrado con RUS, son:

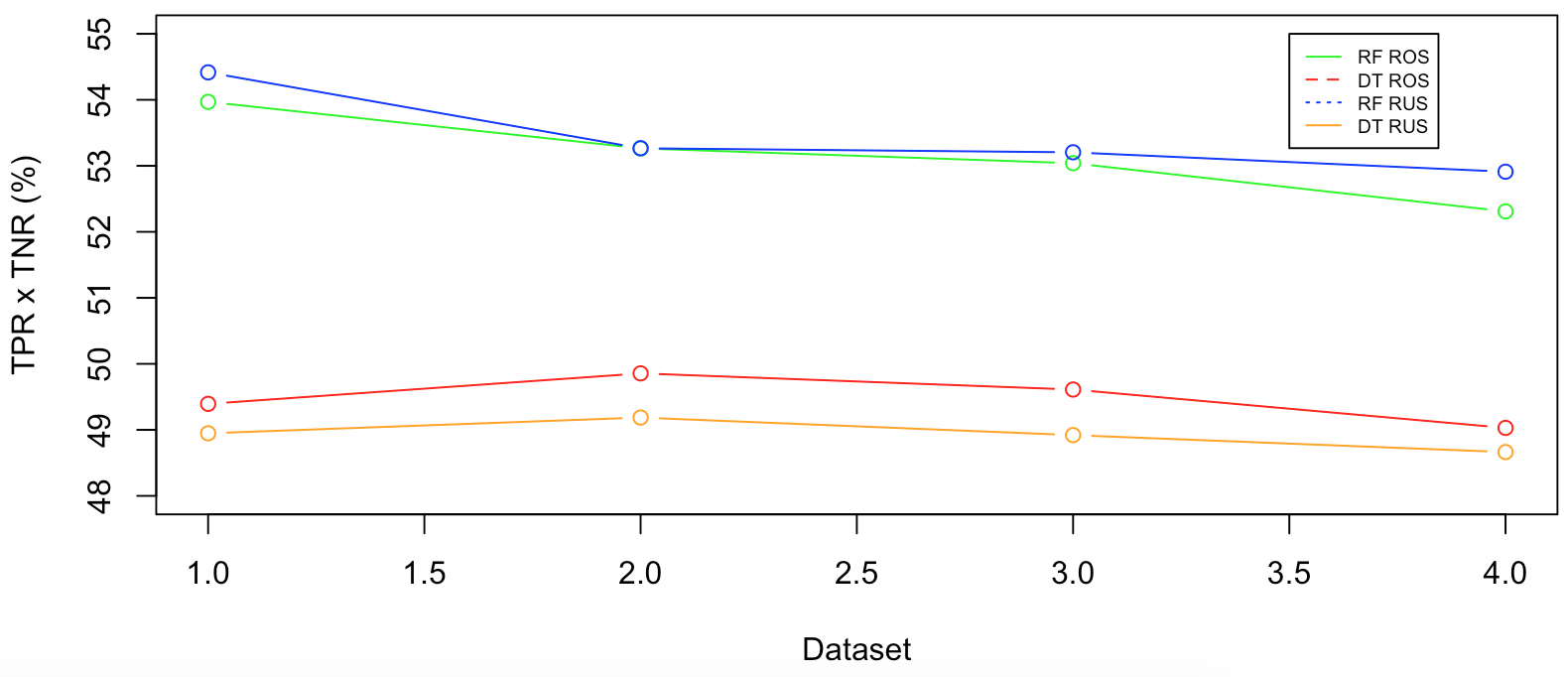
* 1: Selección de 200 características
* 2: Selección de 150 características
* 3: Selección de 125 características
* 4: Selección de 100 características

Obteniendo los siguientes resultados:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datos + Modelo | Resultados | | | | | | | | Tiempo (s) |
| Test Error % | Precisión % | ROC % | TP | FP | FN | TN | TPR x TNR |
| 1 RF | 26,335 | 73,664 | 73,767 | 202077 | 35656 | 73014 | 101889 | 0,5442 | 373 |
| 1 DT | 30,364 | 69,635 | 69,97 | 189719 | 39923 | 85372 | 97622 | 0,4895 | 60 |
| 2 RF | 26,941 | 73,058 | 72,983 | 201390 | 37469 | 73701 | 100076 | 0,5327 | 218 |
| 2 DT | 30,079 | 69,92 | 70,136 | 191160 | 40187 | 83931 | 97358 | 0,4919 | 12 |
| 3 RF | 27,022 | 72,977 | 72,941 | 200951 | 37364 | 74140 | 100181 | 0,5321 | 224 |
| 3 DT | 30,229 | 69,77 | 69,945 | 190971 | 40617 | 84120 | 96928 | 0,4892 | 11 |
| 4 RF | 27,013 | 72,986 | 72,743 | 202114 | 38491 | 72977 | 99054 | 0,5291 | 211 |
| 4 DT | 30,146 | 69,853 | 69,76 | 192671 | 41976 | 82420 | 95569 | 0,4866 | 10 |

## Comparación resultados

Veamos ahora una comparación en modo gráfica:



De modo que, para cada selección de características, tenemos los distintos valores de los modelos, siendo Random Forest el que mejor ha funcionado. Además, la mejor selección de características ha sido la de 200 características, obteniendo un valor de 54,42% en la medida TPR x TNR.

# 7 Conclusiones

Viendo el estudio realizado sobre este dataset, las distintas técnicas de preprocesamiento usadas como oversampling y undersampling y selección de características, podemos ver cómo tanto con el equilibrado, creando instancias o eliminando instancias, proporciona resultados semejantes.

El algoritmo que mejor resultado nos ha dado ha sido Random Forest con un valor del 54,42% en la medida TPR x TNR, con una configuración de 50 árboles y una máxima profundidad de 16. Esto hay que añadirle que se ha realizado una selección de 200 características.

Como conclusión podemos decir que se ha realizado el estudio sobre un dataset de tamaño grande, sobre un supercomputador usando Spark, por lo que es una gran oportunidad de comprobar de primera mano cómo es trabajar en Big Data.