Universidad de Valladolid

Facultad de Ciencias Económicas y Empresariales

Trabajo de Fin de Grado

Grado en economía

"Big data y las Ciencias Sociales"

Presentado por:

Miguel Arnaiz Fernández

Tutelado por:

Ursicino Carrascal Arranz

Valladolid, 23 de julio de 2018

ÍNDICE DE CONTENIDO

ÍNDICE DE CONTENIDO 1

ÍNDICE DE TABLAS, GRÁFICOS, FIGURAS Y CUADROS. 3

RESUMEN DEL TRABAJO. 4

1. INTRODUCCIÓN AL BIG DATA. 5

1.1 Rasgos característicos del Big Data 6

2. ESTUDIO PRÁCTICO: *GIVE ME SOME CREDIT* 8

2.1 Introducción y objetivos 8

2.2 Análisis exploratorio 11

2.3 Clasificación aleatoria 13

2.4 Regresión logística 14

2.5 Random Forest 21

2.6 XGBOOST 24

2.7 Comparación de modelos 26

3. CONCLUSIONES 29

Bibliografía 31

ÍNDICE DE TABLAS, GRÁFICOS, FIGURAS Y CUADROS.

- **Gráfico 1.1: tráfico de datos en internet en 1 minuto. Fuente: Cumulus Media.**
- **Tabla 2.1: tabla de variables de entrada. Fuente: Kaggle.**
- **Gráfico 2.2: División de un dataset. Fuente: Elaboración propia.**
- **Gráfico 2.3: Esquema de trabajo Data Science. Fuente: Elaboración propia.**
- **Tabla 2.2: Resumen estadístico de las variables. Fuente: elaboración propia.**
- **Tabla 2.3: Distribución de ingresos y familia después de suavizar. Fuente: elaboración propia.**
- **Tabla 2.4: Ejemplo de una matriz de confusión. Fuente: elaboración propia**
- **Tabla 2.5: matriz de confusión para el modelo 1. Fuente: elaboración propia.**
- **Tabla 2.6: matriz de confusión para el modelo 2. Fuente: elaboración propia.**
- **Gráfico 2.4: Funcionamiento gráfico de un árbol de decisión. Fuente: elaboración propia.**
- **Gráfico 2.5: Funcionamiento gráfico del Random Forest. Fuente: elaboración propia.**
- **Gráfico 2.6: Importancia de variables en Random Forest. Fuente: elaboración propia.**
- **Gráfico 2.7: ejemplo de curvas ROC. Fuente: DBT ventures.**
- **Gráfico 2.8: curvas ROC de los modelos. Fuente: Elaboración propia.**

RESUMEN/ABSTRACT.

Este TFG se basa en el estudio de las nuevas formas de trabajo con Big Data y Data Science en el mundo de las finanzas.
La primera parte del trabajo analiza las novedades que ha traído el Big Data, así como sus principales características, conocidas como las 5Vs.
En el segundo apartado se ha llevado a cabo un estudio Data Science para contrastar diferentes algoritmos de predicción en un entorno de datos financieros. En este caso en concreto, se intentará predecir si una persona será capaz de devolver un crédito y, por tanto, saber si debemos concedérselo.
Finalmente, se comparan los resultados que hemos obtenido con los distintos algoritmos para conocer cual ha obtenido los mejores resultados.
Palabras clave: Big Data, Data Science, estadística, predicción.

This TFG is based on the study of new ways of working with Big Data and Data Science in the finances world.
The first part of the work analyzes the innovations that Big Data has brought, as well as its main characteristics, known as 5Vs.
In the second section, we carried out a Data Science study to compare different prediction algorithms in a financial data environment. In this case, we will try to predict if a person will be able to return a loan and, therefore, know if we should grant it.
Finally, we compare the results we have obtained with the different algorithms to know which have obtained the best results.
Key words: Big Data, Data Science, statistics, prediction.

1. INTRODUCCIÓN AL BIG DATA.

La importancia que ha adquirido la información en nuestros días es crucial y enormemente significativa. El estar conectado de forma ininterrumpida a internet se ha convertido en una necesidad en nuestras vidas.

No es sólo es que necesitemos una cantidad ingente de GB para almacenar nuestros datos, es que, internet es un foco incesante de generación de información. En sus inicios, internet estaba constituido como un sistema de intercambio de datos entre ordenadores, pero eso cambió con la llegada del Smartphone, el teléfono que permite estar conectado las 24 horas.

La escala de generación de información en internet es tan enorme que no tiene sentido medirla en meses como puede hacerse en un negocio tradicional. Podemos hacernos una idea de la magnitud viendo el siguiente gráfico:

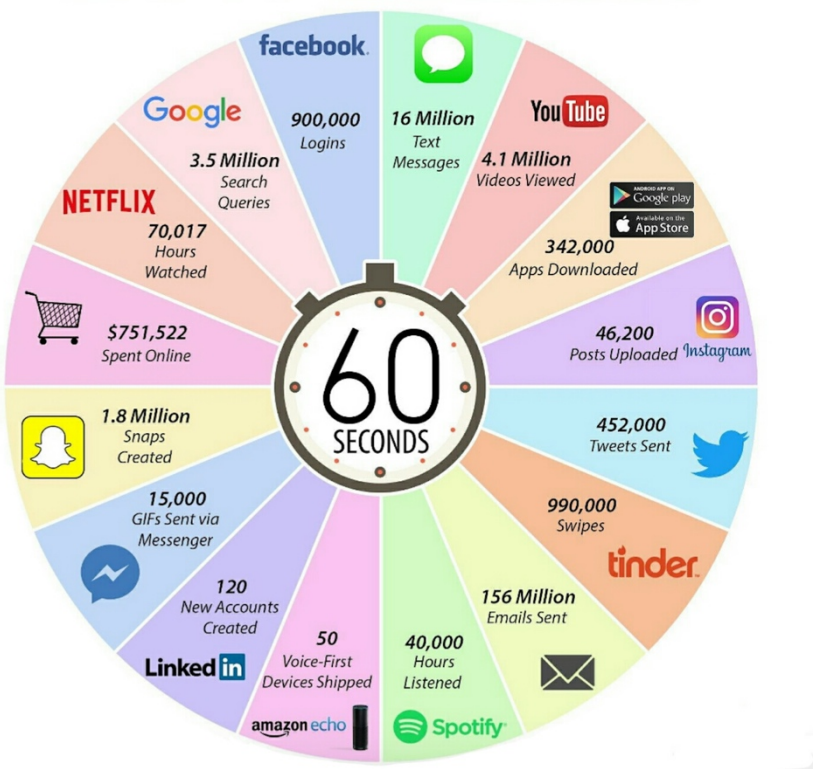


Gráfico 1.1: tráfico de datos en internet en 1 minuto. Fuente: Cumulus Media.

Estas cifras nos hacen darnos cuenta del mundo tan ‘’interconectado’’ en el que vivimos, y ahí es donde entra el Big Data. Desde ya hace varios años es común la utilización de datos para campañas de marketing o para analizar sectores empresariales, pero Big Data ha supuesto una resolución en el trabajo con datos estadísticos. El término Big Data hace referencia un conjunto masivo de datos, que hace imposible su gestión y análisis con las herramientas tradicionales. Otra forma más intuitiva de definir Big Data sería como la utilización de datos para resolver problemas, ya sean de carácter empresarial, personales o del sector público.

1.1 Rasgos característicos del Big Data

Una vez presentados los campos donde se pone en práctica esta herramienta, podemos pasar a estudiar los rasgos característicos y diferenciadores del Big Data frente a otras técnicas de análisis. Estos rasgos suelen ser definidos como las 3 uves:

·**Volumen:** como ya hemos contado, estamos sufriendo un proceso de ‘’datificación de la sociedad’’. Cada vez estamos generando más datos y toda esa información tiene que ser gestionada con nuevas tecnologías. A esto se refiere el volumen, a esta enorme cantidad de datos que se generan. Lo bueno de esto es que las decisiones que se toman son mejores ya que están basadas en una muestra de datos mucho mayor. Por esto contar con estos grandes volúmenes de datos está siendo tan importante en materia de gestión. Un buen ejemplo de esto es que Google gestiona 20 petabytes de datos al día.

·**Velocidad:** la velocidad refleja la frecuencia con la que la información se va generando, para ser almacenada y gestionada. Para muchas tareas, la velocidad es incluso más importante que el propio volumen de datos. Cada día somos testigos de cómo la velocidad de respuesta es crucial en multitud de campos. La ventaja que ofrece Big Data es que, con su velocidad de gestión, permite llevar

estudios de millones de datos a tiempo real, lo que supone una verdadera ventaja comparativa ante modelos de negocio de tradicionales que necesitan varios días para analizar toda esa información. Un ejemplo de la diferenciación que ofrece la velocidad en la gestión de datos es el caso de eBay, que analiza 5 millones de transacciones a través de PayPal al día, en busca de posibles fraudes en su página de compraventa.

·**Variedad**: como ya hemos comentado, cada día utilizamos una amplia variedad de dispositivos conectados a internet que están continuamente generado datos, pero estos tienen formas muy diferentes: mensajes de texto, imágenes publicadas, señales GPS. Estos datos tan dispares no pueden encajarse entre ellos en estructuras sencillas, con un fácil manejo, por lo que se ha tenido que desarrollar un nuevo método de poder utilizarlos. A esto hay que añadir la evolución hacia la reducción de los costes de todos los elementos informáticos de almacenamiento y gestión de la información, lo que ha ayudado a la proliferación de esta tecnología.

Últimamente, además de estas tres características, para definir los rasgos del Big se está extendiendo el uso de otras dos, por lo que pasarían a ser 5 uves:

·**Veracidad:** los datos deben ser veraces y replicables por otro analista con las mismas herramientas. Datos que sean recabados de forma errónea puede dar lugar a conclusiones equivocadas.

·**Valor:** esta característica se refiere a como la acumulación de datos por parte de empresas genera un valor añadido a estas. Cada vez se oyen más noticas de empresas que venden datos a otras o intento de robo de estos por parte de delincuentes informáticos.

Todas estas características que definen el Big Data hacen que se trate de una las mayores revoluciones tecnológicas en lo que análisis de datos se refiere. De la mano de esta tecnología grandes empresas que nacieron como digitales, casos

como Google o Amazon, se han convertido en los pioneros y expertos en el análisis de datos con esta nueva herramienta. La ventaja competitiva la conseguirán las empresas que entiendan mejor lo que está sucediendo y pongan en práctica toda esta nueva ciencia para el análisis y gestión de este volumen de información.

2. ESTUDIO PRÁCTICO: *GIVE ME SOME CREDIT*

Para probar la eficacia que le estamos otorgando en el punto uno al Big Data se ha escogido el ejercicio "Give me some credit". En los siguientes apartados aplicaremos las técnicas propias del Data Science al problema que supone la concesión de créditos; a la hora de otorgar un préstamo uno de los mayores retos es predecir la probabilidad de que el prestatario nos lo vaya a devolver o no. Surge la necesidad, por tanto, de estudiar las variables que afecten a esa probabilidad mediante la búsqueda del modelo que mejor resultado obtenga.

2.1 Introducción y objetivos

Para crear el modelo objetivo contrastaremos los resultados de distintos algoritmos, aplicados sobre un conjunto de datos de carácter financiero.

El proveedor de datos es Kaggle, una plataforma de información y comunidad de data scientist, investigadores y profesionales -adquirida recientemente por Google- que, hace unos años, lanzó el reto de calcular y predecir el impago de préstamos por parte de unos supuestos clientes de una entidad financiera anónima.

Mediante la información proporcionada por el siguiente conjunto de variables crearemos los algoritmos que deben predecir la posibilidad de impago de los clientes

Variable	Carácter	Descripción
SeriousDlqin2yrs	Explicada	Cliente clasificado como moroso por haber excedido 90 días o más de impagos.
RevolvingUtilizationOfUnsecuredLines	Explicativa	Ratio que relaciona el balance de la cuenta de crédito abierta con el límite de crédito establecido.
Age	Explicativa	Edad del cliente.
NumberOfTime30-59DaysPastDueNotWorse	Explicativa	Número de veces que el cliente se ha retrasado en el pago de 30 a 59 días, pero su posición no ha empeorado en los últimos dos años.
DebtRatio	Explicativa	Ratio que relaciona los gastos mensuales (amortización de la deuda y consumo diario) con los ingresos.
MonthlyIncome	Explicativa	Ingresos mensuales.
NumberOfOpenCreditLinesAndLoans	Explicativa	Número de líneas de crédito y préstamos abiertos.
NumberOfTimes90DaysLate	Explicativa	Número de veces que el cliente se ha retrasado en el pago 90 días o más.
NumberRealEstateLoansOrLines	Explicativa	Número de préstamos hipotecarios e inmobiliarios, incluidas líneas de crédito con garantía hipotecaria.
NumberOfTime60-89DaysPastDueNotWorse	Explicativa	Número de veces que el cliente se ha retrasado en el pago de 30 a 59 días, pero su posición no ha empeorado en los últimos dos años.
NumberOfDependents	Explicativa	Número de miembros de la familia del cliente (excluyéndose ellos mismos).

Tabla 2.1: tabla de variables de entrada. Fuente: Kaggle.

Para realizar un contraste entre los distintos algoritmos, primeramente, se comprobará el resultado de aplicar una **función aleatoria** sobre el conjunto de test,

sin ningún tipo de análisis o toma de decisión. En segundo lugar, se estimará una **Regresión Logística**. En tercer lugar, se diseñará un algoritmo **Random Forest**. En cuarto y último lugar, se escogerá el algoritmo **XGBoost**, que al igual que con los anteriores, será entrenado, validado, testado y finalmente evaluado por Kaggle. Toda la explicación sobre cómo funciona cada algoritmo será explicada más adelante en su apartado correspondiente.

La base de trabajo que ofrece Kaggle son dos archivos, correspondientes a la muestra de entrenamiento (training-set, con los resultados de la variable a predecir), y la muestra de test (test-set, con la columna a predecir vacía, la cual, habrá que rellenar con los resultados del algoritmo).

Como explica el siguiente esquema, se ha dividido la muestra de training en dos partes, con el propósito de obtener una muestra de validación que permita comparar entre distintos diseños de algoritmos. De este modo, se evalúa el funcionamiento de cada uno, y se selecciona aquel cuya clasificación sea más acertada.

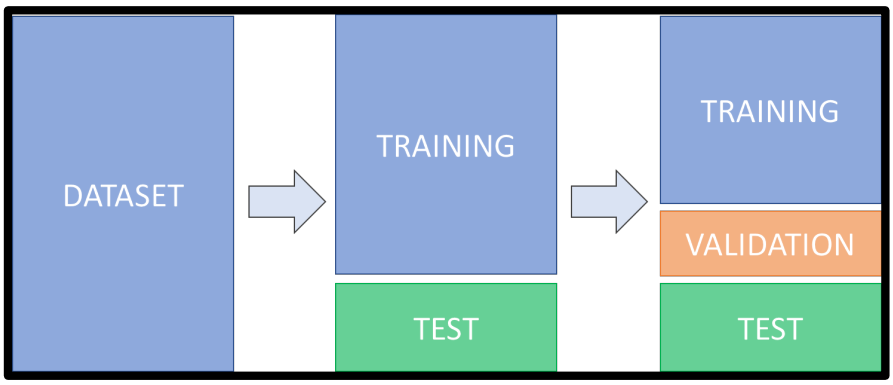


Gráfico 2.2: División de un dataset. Fuente: Elaboración propia.

Por otro lado, el proceso de entrenamiento, validación y testeo en cada uno de los algoritmos será algo standard y queda recogido en el esquema que se expone a continuación. Este modelo es válido para Regresión Logística, Random Forest y XGBoost:

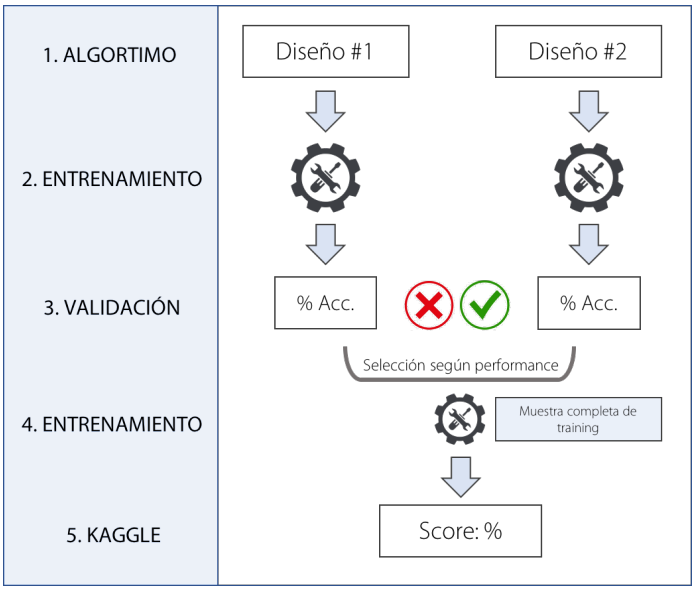


Gráfico 2.3: Esquema de trabajo Data Science. Fuente: Elaboración propia.

Como puede apreciarse, es un sistema de decisión basado en el "performance" de algoritmos. Estos son entrenados bajo una misma muestra de training, y evaluados bajo la misma muestra de validación. Una vez esto ha sido contrastado, el algoritmo que ha proporcionado la mejor repuesta es entrenado con todo el conjunto de training, y realizará la predicción sobre el conjunto de test. Los resultados se devolverán a Kaggle, y de este modo, se conseguirá una puntuación correspondiente a la precisión clasificatoria del modelo.

2.2 Análisis exploratorio

Antes de comenzar a diseñar los algoritmos, es necesario realizar una vista preliminar del resumen estadístico de los datos con los que trabajaremos. Se ha decidido cambiar el nombre de las variables para simplificar el proceso.

Resumen estadístico

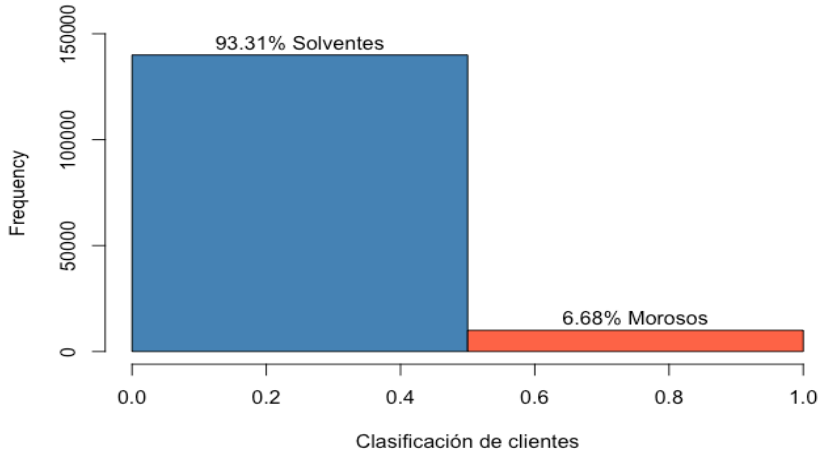
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
Balance/Límite	0	0.03	0.15	6.05	0.56	50708	0
Edad	0	41	52	52.30	63	109	0
Nretrasos	0	0	0.00	0.42	0.00	98	0
Ingresos/Gastos	0	0.18	0.37	353	0.87	329664	0
Ingresos	0	3400	5400	6670	8249	3008750	29731
Ncréditos	0	5	8.00	8.45	11.00	58	0
Nretrasos2	0	0	0	0.27	0	98	0
Npréstamos	0	0	1	1.02	2	54	0
Nretrasos3	0	0	0	0.24	0	98	0
Familia	0	0	0	0.76	1	20	3924

Tabla 2.2: Resumen estadístico de las variables. Fuente: elaboración propia.

Aquí vemos un cuadro resumen con los principales estadísticos de cada una de las variables que vamos a utilizar para el diseño de nuestros modelos.

Observando los valores máximos, destacan algunas variables como *Balance/Límite* e *Ingresos/Gastos*, que, en comparación con la media y la mediana de estas, parecen observaciones muy alejadas del conjunto.

Por último, la tabla muestra las variables con valores ausentes. En este caso concreto, se obtiene que Ingresos tiene 29.731 NAs, y Familia con 3.924 NAs. Estos valores habrán de ser tratados más adelante.



Como puede apreciarse en el histograma, la muestra está muy lejos de poder considerarse balanceada. Esto podría traer problemas en el entrenamiento de algoritmos, ya que, al existir una mayoría de casos clasificados como solventes, es posible que los morosos acaben siendo detectados como ruido. Este aspecto será tratado más adelante.

2.3 Clasificación aleatoria

Pese a no ser estrictamente un algoritmo, resulta interesante realizar una prueba de cuál es la puntación que recibiría una clasificación binaria generada mediante una función aleatoria. Para ello vamos a crear un vector de 101503 registros con una distribución uniforme aleatoria entre 0 y 1.

Comprobamos los resultados subiendo el resultado a Kaggle y obtenemos una puntuación de 0.499% de acierto. Este resultado era previsible ya que se intenta predecir una variable Bernoulli con el “método de la moneda”.

2.4 Regresión logística

La regresión logística se incluye dentro del conjunto de las denominadas técnicas estadísticas del análisis de datos. Su uso se hace imprescindible cuando se quiere relacionar una variable dependiente cualitativa con una o más variables independientes. Resulta útil para los casos en los que se desea predecir la presencia o ausencia de una característica o resultado según los valores de un conjunto de predictores. Es similar a un modelo de regresión lineal, pero está adaptado para modelos en los que la variable dependiente es dicotómica (ceros o unos).

El primer paso es tratar los valores ausentes. Como se ha visto anteriormente existen dos columnas con NAs: Ingresos y Familia, con 29731 y 3924 respectivamente.

En primer lugar, hay que valorar si se pueden eliminar los registros vacíos; claramente esto no es posible, ya que se estaría perdiendo más de un 25% de la información, una cantidad suficientemente significativa como para no valorar esta opción.

Viendo la distribución de estas variables, se halla la existencia de datos extremos que están influyendo sobre el conjunto -como se ha comentado en el análisis exploratorio-, desviando lo valores medios. Por lo tanto, se ha decidido imputar el valor de la mediana, ya que este no se ve distorsionado por estos valores atípicos.

Aquí se ve el nuevo resumen de las dos variables. Han desaparecido todos los valores ausentes y la distribución no se ha visto muy afectada.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Ingresos	0	3912	5400	6437.72	7400	3008750
Familia	0	0	0	0.74	1	20

Tabla 2.3: Distribución de ingresos y familia después de suavizar. Fuente: elaboración propia.

Se estima un primer modelo “exploratorio” con la muestra de entrenamiento, donde la única modificación es omitir la variable X, la cual, es una variable formada por un vector de números ascendentes que se utiliza como índice.

```
## glm(formula = Clasificación ~ . - X, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2613  -0.3906  -0.3141  -0.2531   4.9206
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.345e+00  4.715e-02 -28.522  < 2e-16 ***
## `Balance/Limite`  -7.005e-05  8.429e-05  -0.831  0.40596
## Edad              -2.810e-02  9.299e-04 -30.220  < 2e-16 ***
## Nretrasos          4.984e-01  1.243e-02  40.089  < 2e-16 ***
## `Ingresos/Gastos` -2.885e-05  1.186e-05  -2.433  0.01499 *
## Ingresos          -3.553e-05  3.519e-06 -10.097  < 2e-16 ***
## Ncréditos         -7.880e-03  2.826e-03  -2.789  0.00529 **
## Nretrasos2         4.610e-01  1.692e-02  27.243  < 2e-16 ***
## Npréstamos         7.068e-02  1.179e-02   5.993 2.07e-09 ***
## Nretrasos3        -9.268e-01  1.979e-02 -46.845  < 2e-16 ***
## Familia           9.939e-02  1.015e-02   9.791  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 58556  on 119999  degrees of freedom
## Residual deviance: 53765  on 119989  degrees of freedom
## AIC: 53787
##
## Number of Fisher Scoring iterations: 6
```

Los coeficientes del modelo se muestran en la primera columna de la tabla, bajo el nombre de “estimate”. A continuación, aparecen el error estándar, y el valor z (distr normal) que es el coeficiente dividido por el error.

Se muestran los símbolos “*" y “.”, que indican la significación de los parámetros a diferentes niveles. A un nivel de significación 0,05 los parámetros asociados a las variables con “*" son significativamente distintos de 0, ya que el valor del estadístico de Wald es en valor absoluto mayor que el punto crítico z/2 = 1,96.

Los coeficientes del modelo son:

$$RegLog1 = -1.345 - 7.005e^{-5} * Balance/Límite - 0.0281 * Edad + 0.4984 * Nretrasos - 2.885e^{-5} * Ingresos/Gastos - 3.553e^{-5} * Ingresos - 0.00788 * Ncréditos + 0.461 * Nretrasos2 + 0.07068 * préstamos - 0.9268 * Nretrasos3 + 0.09939 * Familila$$

En cuanto a los coeficientes, la interpretación cambia respecto a un modelo de regresión lineal (lm). El modelo GLM no ajusta la variable respuesta sino una función de enlace. En el caso del modelo logit esta función es:

$$logit[p(x)] = ln[p(x)/1 - p(x)]$$

siendo p la probabilidad de que el individuo tome el valor “1” en la variable dicotómica. Por tanto, para hallar la probabilidad de cada observación es necesario despejar p(x) de la función:

$$p(x) = 1/(1 + e^{-x})$$

Un segundo diseño para otra regresión sería incluyendo variables dummy. Estas variables son utilizadas para representar subgrupos dentro de la muestra. En este caso concreto, se dividen las variables “Edad” y “Ncréditos” en diferentes intervalos, asignado 0 o 1 en función de si pertenece al intervalo o no. Una de las principales ventajas que tiene el uso de variables dummy, es la posibilidad de utilizar una única ecuación de regresión para representar múltiples grupos.

Vemos un ejemplo de que como queda la edad una vez se ha divido en subgrupos aplicándole el uso de variables dummy.

```
  Edad(-Inf,20) Edad(20,30) Edad(30,45) Edad(45,60) Edad(60,75)
1             0           0           1           0           0
2             0           0           1           0           0
3             0           0           1           0           0
4             0           1           0           0           0
5             0           0           0           1           0
6             0           0           0           0           1
  Edad(75,90)   Edad(90, Inf]
1           0             0
2           0             0
3           0             0
4           0             0
5           0             0
6           0             0
```

Vamos a estimar el segundo modelo con los datos divididos en subgrupos.

```
## Call:
## glm(formula = Clasificación ~ . - X - `Edad(90, Inf]` - `Ncréditos(40, Inf]`,
##     family = "binomial", data = trainRL)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1935  -0.3846  -0.3239  -0.2335   5.1026
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.858e+00  5.825e-01  -4.906 9.31e-07 ***
## `Balance/Limite`    -6.631e-05  8.465e-05  -0.783 0.433435
## Nretrasos            5.011e-01  1.243e-02  40.326  < 2e-16 ***
## `Ingresos/Gastos`   -3.184e-05  1.187e-05  -2.682 0.007311 **
## Ingresos            -3.891e-05  3.578e-06 -10.874  < 2e-16 ***
## Nretrasos2           4.340e-01  1.679e-02  25.841  < 2e-16 ***
## Npréstamos           6.681e-02  1.174e-02   5.690 1.27e-08 ***
## Nretrasos3          -9.049e-01  1.969e-02 -45.952  < 2e-16 ***
## Familia              9.265e-02  1.050e-02   8.825  < 2e-16 ***
## `Edad(-Inf,20]`     -6.387e+00  7.246e+01  -0.088 0.929766
## `Edad(20,30]`        1.256e+00  3.235e-01   3.883 0.000103 ***
## `Edad(30,45]`        1.170e+00  3.224e-01   3.629 0.000284 ***
## `Edad(45,60]`        9.051e-01  3.224e-01   2.808 0.004988 **
## `Edad(60,75]`        2.435e-01  3.234e-01   0.753 0.451511
## `Edad(75,90]`       -2.103e-01  3.315e-01  -0.634 0.525840
## `Ncréditos(-Inf,3]` -3.571e-01  4.861e-01  -0.735 0.462624
## `Ncréditos(3,9]`    -9.079e-01  4.849e-01  -1.872 0.061181 .
```

```
## `Ncréditos(9,20]`   -8.092e-01  4.844e-01  -1.671 0.094812 .
## `Ncréditos(20,29]`  -5.785e-01  4.891e-01  -1.183 0.236874
## `Ncréditos(29,40]`  -7.278e-01  5.430e-01  -1.340 0.180141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 58556  on 119999  degrees of freedom
## Residual deviance: 53544  on 119980  degrees of freedom
## AIC: 53584
##
## Number of Fisher Scoring iterations: 8
```

Aquí se ve el resumen de este segundo modelo. Una forma rápida de estimar que modelo es mejor es comparar su AIC. El criterio de información de Akaike (AIC) es una medida de la calidad relativa de un modelo estadístico, para un conjunto dado de datos. Como tal, el AIC proporciona un medio para la selección del modelo.

La fórmula para calcular el AIC es:

$$AIC = 2k - 2ln(L)$$

donde k es el número de parámetros en el modelo y L es el máximo valor de la función de verosimilitud para el modelo estimado.

Vemos que en el segundo modelo es algo inferior, 53787 y 53584 respectivamente. Esto no es muy reseñable así que la mejor forma de ver qué modelo es mejor es ver sus marices de confusión.

Una matriz de confusión tiene la siguiente estructura:

		PREDICCIÓN	
		NEGATIVOS	POSITIVOS
REAL	NEGATIVOS	VERDADEROS NEGATIVOS (VN)	FALSOS NEGATIVOS (FP)
	POSITIVOS	FALSOS NEGATIVOS (FP)	VERDADEROS POSITIVOS (VN)

Tabla 2.4: Ejemplo de una matriz de confusión. Fuente: elaboración propia

- **VP** es la cantidad de *positivos* que fueron *clasificados correctamente* como positivos por el modelo.
- **VN** es la cantidad de *negativos* que fueron *clasificados correctamente* como negativos por el modelo.
- **FN** es la cantidad de *positivos* que fueron *clasificados incorrectamente* como negativos.
- **FP** es la cantidad de *negativos* que fueron *clasificados incorrectamente* como positivos.

Para conocer la precisión de un modelo basta con sumar la diagonal principal de la matriz y dividirla por la suma total.

Aquí tenemos las matrices de confusión tanto del modelo 1 como del modelo 2 con las variables dummy.

MODELO 1		PREDICCIÓN	
		0	1
REAL	0	27870	61
	1	1985	84

Tabla 2.5: matriz de confusión para el modelo 1. Fuente: elaboración propia.

MODELO 2		PREDICCIÓN	
		0	1
REAL	0	27869	62
	1	1976	93

Tabla 2.6: matriz de confusión para el modelo 2. Fuente: elaboración propia.

Pese a conseguir unos buenos resultados en los modelos (93% y 93,3% respectivamente), tenemos que subir la solución final a la plataforma Kaggle para ver el resultado final.

Algortimo	**Dummys**	**Kaggle Score**
GLM	Sí	0.709114

Puede parecer un poco ilógico que arroje una puntuación tan baja, pero hay que tener en cuenta que estamos clasificando posibles impagos de préstamo. En este caso concreto es evidente que tenemos que dar más importancia al falso negativo que al falso positivo, ya que estaríamos concediendo créditos a personas que en principio no disponen de recursos para hacerlos frente.

Vemos la muestra que hemos utilizado para la validación del modelo y presenta la siguiente distribución:

0	**1**
27931	2069

Como se ve, hay alrededor de 2000 personas clasificadas como morosos y el mejor algoritmo sólo ha conseguido clasificar a 93 bien. Esto hace que la precisión

matemática del modelo sea superior al 93%, pero en la realidad está lejos de llegar a esas cotas.

Podemos concluir con que la regresión logística es uno de los modelos más sencillos en el ámbito de la predicción y clasificación. Suele utilizarse como base de cara a una investigación más exhaustiva. El hecho de haber obtenido una puntuación superior al 70%, demuestra un buen funcionamiento tanto del algoritmo como del uso de variables dummy, aunque necesitamos una mejora de la clasificación de los casos catalogados como morosos.

2.5 Random Forest

Random Forest es un algoritmo derivado de los árboles de decisión estándar. Su esencia, principalmente, se encuentra en el uso de un número determinado de árboles de decisión, junto a un parámetro aleatorio de selección de variables a la hora de realizar particiones en cada nodo. A diferencia de un solo árbol de decisión, este algoritmo realiza una votación del resultado de cada árbol del "bosque", obteniendo así -dada una nueva observación- la predicción estimada.

Simplificando, se podría decir que: “El Random Forest construye varios árboles de decisión y los combina para conseguir una predicción más acertada” (Niklas Donges, 2018). Este método es muy preciso ya que evalúa diferentes opiniones acerca del resultado de la clasificación. Además, el parámetro aleatorio de selección de variables ayuda a que no se produzca overfitting o sobreajuste del modelo. En cada nodo del árbol, se escogen n número de variables de forma aleatoria, y aquella que garantice la mejor partición clasificatoria, será la usada. En contraposición, un árbol de decisión tendría acceso a todas las variables en cada nodo, y estaría ajustándose en exceso a la muestra de training.

La diferencia fundamental se encuentra en la forma de clasificar. Gráficamente, se podría resumir de la siguiente forma:

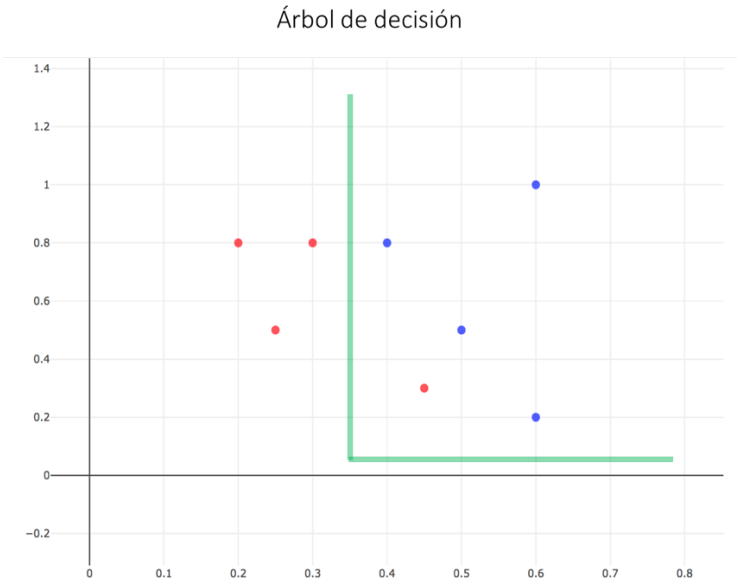


Gráfico 2.4: Funcionamiento gráfico de un árbol de decisión. Fuente: elaboración propia.

Existe una observación que ha quedado mal clasificada por la rigidez del árbol de decisión. (Hay que decir que los datos no son reales, es sólo una representación con fines explicativos).

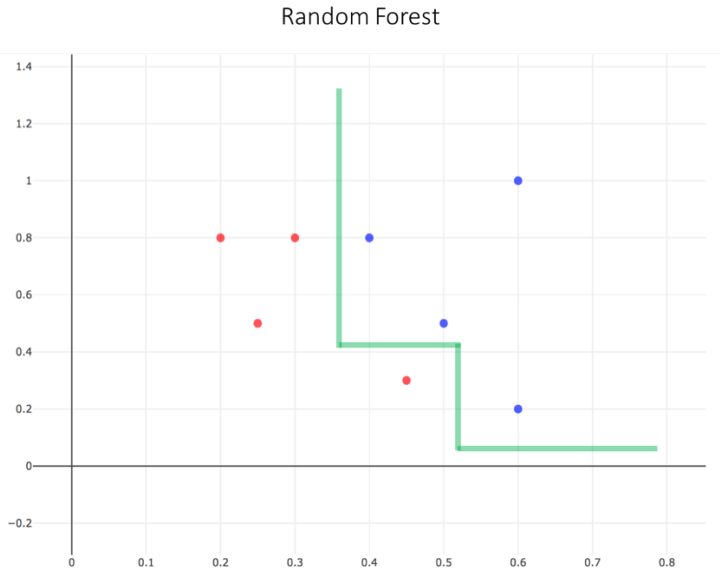


Gráfico 2.5: Funcionamiento gráfico del Rando Forest. Fuente: elaboración propia.

En este caso, el Random Forest es capaz de clasificar adecuadamente una observación que dista del resto de su grupo, gracias a la evaluación conjunta de varios árboles y el parámetro aleatorio que selecciona variables.

Una de las ventajas que ofrece el Random Forest es que permite depurar el modelo que se está diseñando puntuando la importancia que tiene cada variable a la hora de clasificar. De esta forma, podemos obtener un modelo más sencillo y en muchos casos que obtenga mejores resultados.
Haciendo el estudio de importancia de variables para el modelo obtenemos los siguientes resultados:

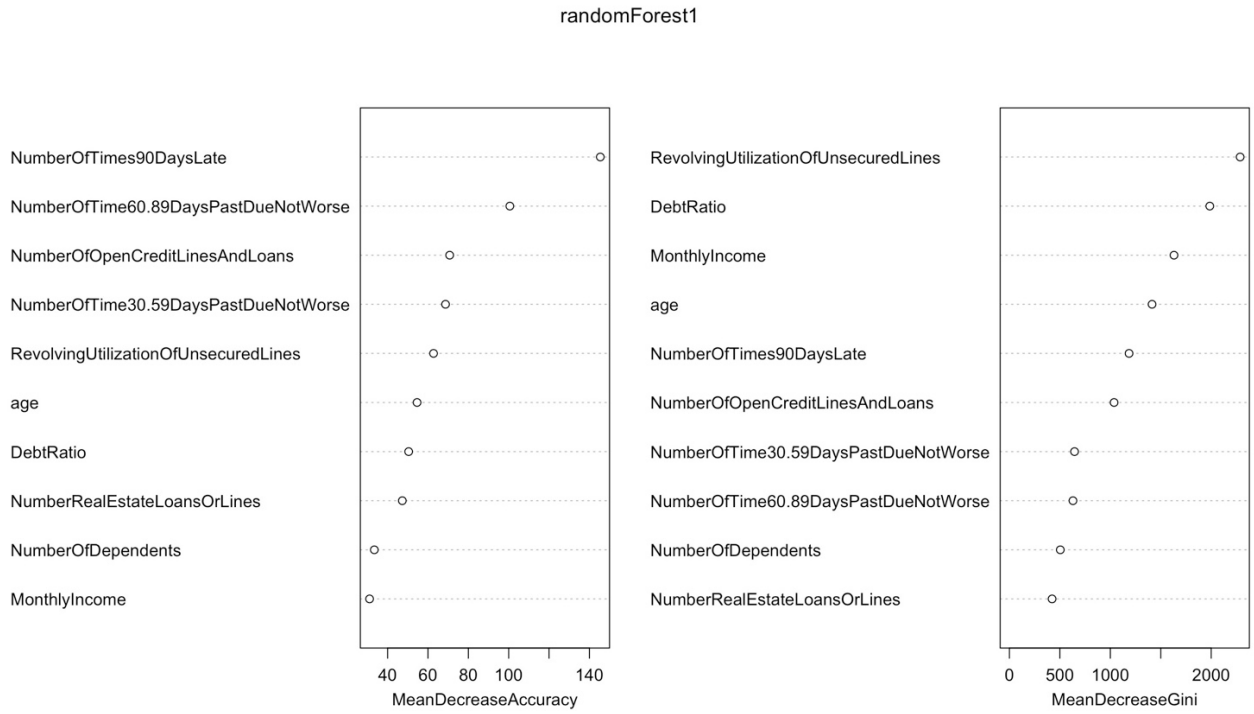


Gráfico 2.6: Importancia de variables en Random Forest. Fuente: elaboración propia.

Las variables Ingresos (MonthlyIncome) y Familia (NumberOfDependents) no aportan demasiado a la precisión del modelo en comparación con el resto. Por ello, se tomó la decisión de omitirlas para el diseño del algoritmo.

Ahora vamos a poner en marcha la predicción utilizando el método de Random Forest. Hemos eliminado de la muestra de entrenamiento las dos variables que menos información aportaban. Además, podemos elegir el tamaño del "bosque" que vamos a utilizar. En este caso serán 500 árboles de decisión.
Una vez ejecutado el algoritmo estos son los resultados que arroja:

Algoritmo	**Número variables**	**Porcentaje de acierto**
Random Forest	9	92.67%

Aunque este resultado es muy alto, tenemos que tener en cuenta que este no es el resultado final, ya que ahora tendríamos que poner nuestro modelo en producción; en este caso, consiste en subir los resultados obtenidos a Kaggle.

Los resultados han sido los siguientes:

Algoritmo	**Variables**	**Kaggle Score**
RF	9	0. 859834

El resultado es muy satisfactorio considerando que la máxima puntuación en Kaggle es de 0.869558, es decir, 0.01 más. Sin embargo, el verdadero reto se encuentra en superar estas centésimas. Hay que destacar los buenos resultados que consigue este algoritmo y su relativa sencillez, mucho mayor que las de otros como las redes neuronales, que en este tipo de problemas conseguirían resultados similares siendo mucho más complejas tanto de diseñar como de entender.

2.6 XGBOOST

XGBoost viene del acortamiento de "eXtreme Gradient Boosting". Vamos a empezar definiendo que es "Boosting". "El término "Boosting" se refiere a la familia de meta-

algoritmos que convierten un conjunto de clasificadores débiles en un clasificador robusto" (Sunil Ray, 2015).

Un ejemplo para poder entenderlo sería la clasificación de emails en SPAM o no:

- Si un email sólo contiene enlaces, probablemente sea SPAM.
- Si un email viene de un usuario conocido no tendría por qué ser SPAM.
- Si un email comienza con: "Ha sido usted seleccionado para ganar…", posiblemente sea SPAM.

Estos tres ejemplos son clasificadores débiles. Por sí mismos no son capaces de distinguir entre SPAM o no, pero una combinación de estas reglas si conseguiría unos buenos resultados en la clasificación de correos. Este es el funcionamiento del algoritmo; lleva a cabo un número determinado de iteraciones sobre los datos de entrenamiento en busca de clasificadores y los va recopilando para construir un clasificador robusto.

Por su parte, el XGboost no es más que un método optimizado y flexible para resolver problemas de ciencia de datos utilizando la técnica boosting. Una ventaja que ofrece XGboost es que es una librería creada por la comunidad, por lo que al ser código libre está en constante mejora y revisión. Este es uno de los motivos por los que dicha técnica está muy de moda, además de que ha sido la ganadora de multitud de competiciones de Data Science y Machine Learninig.

Una vez hemos explicado el funcionamiento de esta técnica sólo nos queda ejecutarlo. Cabe destacar que hemos configurado el algoritmo para que realice 90 iteraciones sobre los datos y así "aprenda" y saque los clasificadores.

En resumen, vamos a ejecutar un algoritmo de tipo XGBoost para todas las variables de la muestra de train de nuestros datos. Este va a realizar 90 iteraciones sobre el dataset, y en cada una de ellas irá encontrando reglas que le permita predecir que observaciones serán clasificadas como morosos y cuáles no.

Una vez entrenado vamos a ver su rendimiento en la muestra de test.

Algoritmo	Número iteraciones	Porcentaje de acierto
XGBoost	90	91.52%

El resultado se encuentra en el rango de acierto del Random Forest. Dado el éxito de este algoritmo, tenemos buenas expectativas acerca del funcionamiento del XGBoost bajo la muestra del test de Kaggle.

Algoritmo	Iteraciones	Kaggle Score
Xgboost	90	0.858913

Los resultados son muy prometedores. Podemos destacar dos puntos muy positivos sobre este algoritmo:

Podemos decir que este algoritmo es muy consistente. Aunque sólo presentamos un diseño, hemos realizado varias pruebas con diferentes hiperparámetros, consiguiendo una capacidad predictiva similar. Esto refleja que el éxito radica en la técnica del “boosting” y no en el algoritmo en sí.

En segundo lugar, hay que destacar que ha sido el algoritmo que mejores resultados ha obtenido sin apenas realizar algún tratamiento a las variables.

2.7 Comparación de modelos

Una vez hemos ejecutado los modelos es hora de comparar sus resultados. Para ello vamos a utilizar un gráfico denominado curva ROC (acrónimo de Receiver Operating Characteristic). Se trata de una herramienta estadística que se utiliza para valorar la eficacia de un algoritmo de clasificación dicotómica. Una posible forma de

interpretar esta curva es la relación que existe entre la ratio de falsos positivos y el de verdaderos positivos. “La curva ROC nos dice lo bien que un modelo es capaz de diferenciar entre dos cosas (por ejemplo, entre si un paciente padece una enfermedad o no).” (Jocelyn D’Souza, 2018). Además, para medir la precisión del algoritmo, utilizaremos el AUC (Area Under the Curve) de cada modelo. El valor del AUC oscila entre 0 y 1, siendo 0 un modelo con todas las predicciones erróneas, y 1 un modelo con todas las predicciones correctas.

A continuación, vamos a ver un ejemplo de curva ROC con 4 métodos de trabajo. Como se ha mencionado anteriormente, el mejor método de clasificación será el que tenga más área por debajo de la curva, en este caso el método 4. A partir de este, los modelos van disminuyendo en predicciones correctas, hasta llegar al método 1, que podría clasificarse como pobre. Todas las curvas se comparan con una línea diagonal que une la esquina inferior derecha con la esquina superior derecha. Esta línea se la conoce como línea de no discriminación o adivinación aleatoria.

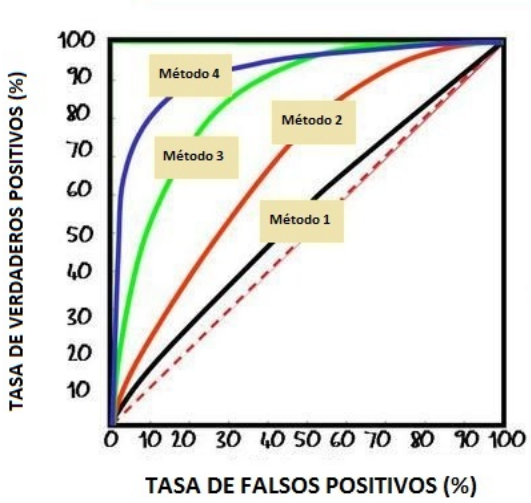


Gráfico 2.7: ejemplo de curvas ROC. Fuente: DBT ventures.

Vamos a ver la curva ROC de nuestros algoritmos representadas por colores. Rápidamente vemos que el que mejor tasa de verdadero positivo y falso positivo obtiene es el Random Forest. En segundo lugar, quedaría el XGBoost y por último la regresión. Además, vemos la línea de no discriminación, es decir, la línea que representa un modelo aleatorio. En este caso nos sirve para comparar los algoritmos que hemos diseñado con el modelo aleatorio. Además, hay que tener en cuenta que esta curva representa los resultados obtenidos en nuestra muestra de test, pero no son los resultados finales del trabajo. Lo que de verdad cuenta es la puntuación que nos devuelve kaggle cuando subimos el modelo. Aun así, siempre hay que llevar a cabo una prueba de test para ver que el modelo funcione correctamente y hacernos una idea de su rendimiento.

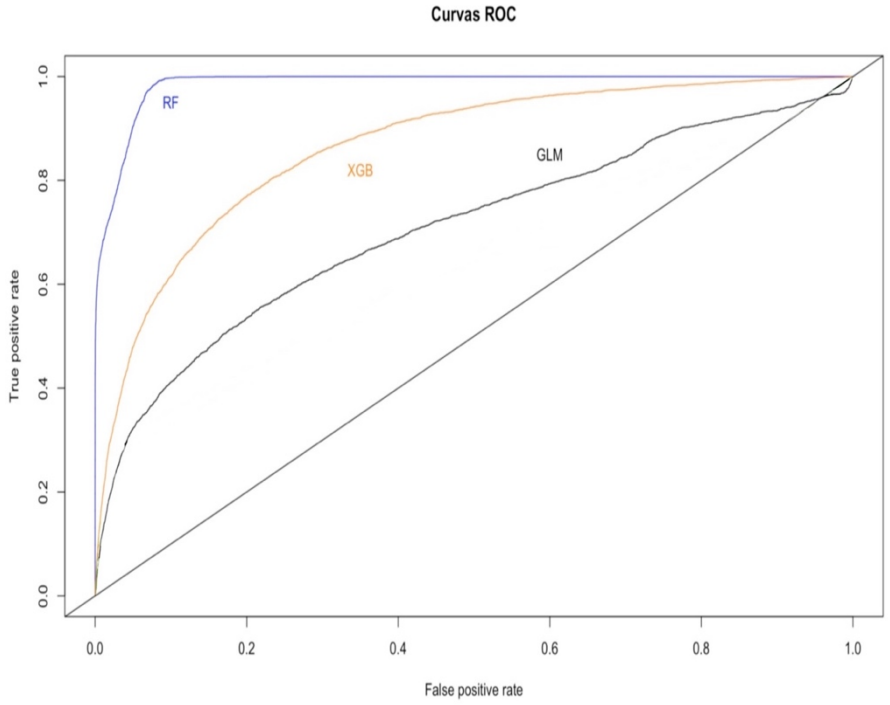


Gráfico 2.8: curvas ROC de los modelos. Fuente: Elaboración propia.

Vamos a ver ahora una tabla con la precisión de cada modelo y su score en Kaggle:

Algoritmo	AUC	KAGGLE SCORE
Regresión Logística	0.706	0.709114
Random Forest	0.986	0. 859834
XGBoost	0.865	0.858913

3. CONCLUSIONES

El propósito de este apartado es resumir los resultados obtenidos anteriormente y calificar los distintos algoritmos.
En el apartado anterior se pueden identificar las distintas curvas ROC de los algoritmos, Intuitivamente, el algoritmo que mejor tasa de falso positivo y verdadero positivo tienen es el Random Forest, seguido por el XGBoost y por último la Regresión Logística.

Tenemos que tener en cuenta que la puntuación obtenida en Kaggle es muy parecida entre el Random Forest y el XGBoost, aunque en nuestra curva ROC se vean tan diferentes. Esto seguramente sea por la muestra de test que hemos utilizado.

Hay que valorar la Regresión Logística su rapidez y validez como estudio preliminar. En el apartado dedicado a este modelo, se han destacado positivamente los resultados obtenidos en Kaggle, que pese a estar basado en una regresión común, eran muy satisfactorios. Sin embargo, la probabilidad de clasificar bien nuevas observaciones -medido por el AUC- no es muy alta, y por ello no puede competir con algoritmos más complejos.

Para concluir, podemos decir que el algoritmo Random Forest es el que mejor relación de eficacia y eficiencia tiene. Esto no significa que en todos los casos sea el más apropiado. Como ha podido demostrarse, tanto el tratamiento de datos como el ajuste del algoritmo son factores claves de cara a realizar un procedimiento de

predicción o clasificación. El XGBoost ofrece unos resultados muy prometedores, tanto en tiempo de computación como en scoring, sin embargo, no ha terminado de superar al famoso Random Forest, ya que este consigue la mejor puntuación de todos los algoritmos y además la probabilidad de clasifica correctamente nuevas observaciones es cercana al 100%.

4. BIBLIOGRAFÍA

- Aldana Montes J. Francisco, Baldominos Gomez Alejandro, Garcia Nieto Jose Manuel, Mochó Morcillo Francisco, Gonzalez CAbañas J. Carlos, Navas Delgado Ismael (2016): Introducción al Big Data. Editorial: García-Maroto

- BS Businnes school: "En 2020, más de 30 mil millones de dispositivos estarán conectados a Internet" (2015) Disponible en: https://www.obs-edu.com/es/noticias/estudio-obs/en-2020-mas-de-30-mil-millones-de-dispositivos-estaran-conectados-internet

- Caballero, Rafael (2017): "Con V de Big Data". El País. Disponible en: https://elpais.com/tecnologia/2017/04/26/actualidad/1493195037_932452.html

- Clasificación: ROC y AUC. Curso intensivo de aprendizaje automático. Google Developers. Disponible en: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=es-419

- Data school (2014) : Simply guide to confusion matrix terminology. Disponible en: http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

- De la Fuente Fernández, Santiago (2011): Regresión Logística. Facultado de CCEE y empresariales UAM. Disponible en: http://www.estadistica.net/ECONOMETRIA/CUALITATIVAS/LOGISTICA/regresion-logistica.pdf

- Donges, Niklas (2018) : "The Random Forest Algorithm ". Towards Data Science (Medium). Disponible en: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

- Gabriel Tseng (2018) : “Gradient Boosting and XGBoost”. Disponible en : https://medium.com/@gabrieltseng/gradient-boosting-and-xgboost-c306c1bcfaf5
- Jocelyn D’Souza (2018) : “Let’s learn about AUC ROC curve!”. Disponible en: https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152
- Medina Moral, Eva (2003): Modelos de elección discreta. Disponible en: http://www.uam.es/personal_pdi/economicas/eva/pdf/logit.pdf
- R documentation “random forest” (2018). Disponible en: https://cran.r-project.org/web/packages/randomForest/randomForest.pdf
- R documentation: Fitting Generalizad Lienar Models. Disponible en: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html
- R documentation: Package “pRoc” (2018). Disponible en: https://cran.r-project.org/web/packages/pROC/pROC.pdf
- ROC your world : the importance of statisticians in SAAS. Disponible en: http://www.dbtventures.com/blog/2015/10/11/roc-your-world
- Sunil Ray (2015) : “Quick introduction to Boosting algorithms in Machine Learning “. Disponible en: https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/