Universidad de Granada

Máster Universitario en Ingeniería Informática

Competición en Kaggle sobre Clasificación Binaria: "Titanic"

Sistemas Inteligentes para la Gestión en la Empresa

Alumno

Marvin Matías Agüero Torales

maguero@correo.ugr.es

Titanic	
Usuario	https://www.kaggle.com/mmaguero
Equipo	mmaguero
Ranking global	202 (Top 4%)
Puntuación	0,82297

2016-17

Contenido

Introducción....3
Exploración y Preprocesamiento de datos....3
Técnicas de clasificación....15
Presentación y discusión de resultados....16
Conclusiones....16
    Trabajo Futuro....16
Listado de soluciones....17
Anexos....19

Introducción

El hundimiento de *Titanic* en el siglo XX es una tragedia sensacional, en la que 1502 de 2224 pasajeros y miembros de la tripulación fallecieron. Kaggle[1] proporcionó este conjunto de datos a principiantes de aprendizaje de máquina para predecir qué tipo de personas tenían más probabilidades de sobrevivir dado la información incluyendo sexo, edad, nombre, etc. Aunque el tamaño del conjunto de datos es pequeño, proporciona a los novatos la oportunidad de lograr la mayoría de los procedimientos en un proyecto de ciencias de datos, que incluye limpieza de datos, ingeniería de funciones, ajuste de modelos y ajustes para lograr puntuaciones más altas (Xiang, 2016).

Exploración y Preprocesamiento de datos

La sección de exploración de datos no la quise separar de la de preprocesamiento, puesto que ambas acciones las iba intercalando, quedando muy entrelazadas, con cada hipótesis hacia un poco de exploración y otro poco de preprocesamiento.

Vemos que hay 891 filas en el conjunto de entrenamiento con 12 variables cada una. El conjunto de pruebas es más pequeño, con sólo 418 pasajeros (a predecir), y sólo 11 variables ya que la columna "Survived" falta predecir (Tabla 1).

```
                                    Entrenamiento
'data.frame':   891 obs. of  12 variables
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : Factor w/ 681 levels "110152","110413",..: 525 596 662 50 ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 ...
 $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...


                                        Prueba
'data.frame':   418 obs. of  11 variables
 $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
 $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
 $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
 $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
 $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
 $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
 $ Ticket     : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 ...
 $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
 $ Cabin      : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

Tabla 1. Tipos de variables.

Podemos ver que la variable de nombre tiene 891 niveles, lo que significa que no hay dos pasajeros comparten el mismo nivel de factor que el número total de filas. Para número de billete y la cabina, hay menos niveles, probablemente porque faltan valores allí (Stephens, 2014). Vemos que 342 pasajeros sobrevivieron y 542 no.

El desastre del *Titanic* es famoso por salvar "las mujeres y los niños primero", y con las variables Sexo y Edad podemos ver que los patrones son evidentes. Así vemos que la mayoría de los

1 https://www.kaggle.com/

pasajeros eran hombres (577 contra 314 mujeres). En la Tabla 2 se puede ver la proporción en fila, es decir, la proporción de cada sexo que sobrevivió, como grupos separados. La mayoría de las mujeres a bordo sobrevivieron (y un porcentaje muy bajo de hombres). Lo primero que debemos hacer es la adición de la columna de predicción "todos mueren", excepto mujeres. (Stephens, 2014).

Surviver	No	Yes
female	0.2579618	0.7420382
male	0.8110919	0.1889081

Tabla 2. Sobrevivientes por sexo.

En la Tabla 3, empezamos a cavar en la variable de edad.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.42	20.12	28.00	29.70	38.00	80.00	177 (263)

Tabla 3. Estadísticas de la variable edad.

Por ahora podríamos suponer que los 177 valores faltantes son la edad media del resto de los pasajeros. Cabe destacar, que los valores falten en el análisis de datos, puede causar una variedad de problemas en el mundo real que puede ser muy difícil tratar a veces. Para categorizar la Edad, vamos a crear una nueva variable binaria (0/No, 1/Si), "Niño", para indicar si el pasajero es menor de 18 años (a cualquier pasajero con una edad de *NA* se le asigna un cero) (Stephens, 2014).

Podemos seguir escarbando con sexo y edad para ver las proporciones de supervivencia para los diferentes subconjuntos, como se ve en la Tabla 4, el número de sobrevivientes para los diferentes subconjuntos. Se subdivide el conjunto de datos sobre las diferentes combinaciones posibles de las variables de edad y sexo, se aplica la función de suma al vector Survived para cada uno de estos subconjuntos. Como nuestra variable objetivo se codifica como 1 para sobrevivir, y 0 para no, el resultado de la suma es el número de supervivientes, y las proporciones se obtienen tomando el vector del subconjunto como entrada, aplicando los comandos de suma y longitud, y luego se hace la división para darnos una proporción (Stephens, 2014).

Child	Sex	Survived
0	female	195
1	female	38
0	male	86
1	male	23

Child	Sex	Survived
0	female	259
1	female	55
0	male	519
1	male	58

Child	Sex	Survived
0	female	0.7528958
1	female	0.6909091
0	male	0.1657033
1	male	0.3965517

Tabla 4. Sobrevivientes por Sexo y edad.

En la Tabla 4, se ve que si un pasajero es mujer sobrevive, independientemente de si eran un niño o no. Pero que pasa si reduzco la mayoría de edad, y me centro en los niños pequeños (menores a 4). En la Tabla 5 se ven los porcentajes de sobrevivientes. Aquí vemos que los hombres solo si eran niños pequeños (menores a 4 años) tenían incluso más posibilidad de sobrevivir que las mujeres (Fila a). Incluso vemos que los niños sobreviven más entre los años 2 y 4, y las niñas no (Fila b).

a	Child	Sex	Survived
	0	female	0.7483444
	1	female	0.5833333
	0	male	0.1717352
	1	male	0.7222222

b	Child	Sex	Survived
	0	female	0.7467949
	1	female	0.0000000
	0	male	0.1867365
	1	male	0.5000000

Tabla 5. Sobrevivientes por sexo y edad (niños menores a 4 años).

Ahora podemos echar un vistazo a un par de otras variables más, potencialmente interesantes, para ver si podemos encontrar algo más: la clase en que estaban, y lo que pagaron por su boleto. La variable de clase se limita a 3 valores (1º a 3º), la tarifa es de nuevo una variable continua que necesita ser reducida a algo que pueda ser fácilmente tabulado: hasta $10, entre $10 y $20, $20 a $30 y más de $30 (Stephens, 2014). Esto podemos verlo en la Tabla 6.

	Fare2	Pclass	Sex	Survived
1	20-30	1	female	0.8333333
2	30+	1	female	0.9772727
3	10-20	2	female	0.9142857
4	20-30	2	female	0.9000000
5	30+	2	female	1.0000000
6	<10	3	female	0.5937500
7	10-20	3	female	0.5813953
8	20-30	3	female	0.3333333
9	30+	3	female	0.1250000
10	<10	1	male	0.0000000
11	20-30	1	male	0.4000000
12	30+	1	male	0.3837209
13	<10	2	male	0.0000000
14	10-20	2	male	0.1587302
15	20-30	2	male	0.1600000
16	30+	2	male	0.2142857
17	<10	3	male	0.1115385
18	10-20	3	male	0.2368421
19	20-30	3	male	0.1250000
20	30+	3	male	0.2400000

Tabla 6. Distribución de sobrevivientes por sexo, edad, clase y tarifa.

Podemos extender el análisis anterior a los niños pequeños (ver Tabla 7). No parece muy determinante agregando clase y tarifa.

Aunque la mayoría de los hombres, independientemente de la clase o la tarifa todavía no sobreviven del todo, nos damos cuenta de que la mayoría de las mujeres de la clase 3 que pagaron más de $20 por su boleto, en realidad también no salvan su vida, quizás las cabinas más caras se situaron cerca del sitio de impacto del iceberg, o más lejos de las escaleras de salida, esto altera nuestras predicciones (Stephens, 2014).

	Fare2	Age2	Pclass	Sex	Survived
1	30+	2-4	1	female	0.00000000
2	20-30	4+	1	female	0.83333333
3	30+	4+	1	female	0.98850575
4	20-30	2-4	2	female	1.00000000
5	30+	2-4	2	female	1.00000000
6	10-20	4+	2	female	0.91428571
7	20-30	4+	2	female	0.89655172
8	30+	4+	2	female	1.00000000
9	10-20	<2	3	female	1.00000000
10	10-20	2-4	3	female	0.50000000
11	20-30	2-4	3	female	0.00000000
12	30+	2-4	3	female	0.00000000
13	<10	4+	3	female	0.59375000
14	10-20	4+	3	female	0.54054054
15	20-30	4+	3	female	0.36842105
16	30+	4+	3	female	0.13333333
17	30+	<2	1	male	1.00000000
18	<10	4+	1	male	0.00000000
19	20-30	4+	1	male	0.40000000
20	30+	4+	1	male	0.37647059
21	10-20	<2	2	male	1.00000000
22	20-30	<2	2	male	1.00000000
23	30+	<2	2	male	1.00000000
24	10-20	2-4	2	male	1.00000000
25	20-30	2-4	2	male	1.00000000
26	<10	4+	2	male	0.00000000
27	10-20	4+	2	male	0.11666667
28	20-30	4+	2	male	0.04545455
29	30+	4+	2	male	0.08333333
30	<10	<2	3	male	1.00000000
31	20-30	<2	3	male	1.00000000
32	30+	<2	3	male	0.00000000
33	10-20	2-4	3	male	1.00000000
34	20-30	2-4	3	male	0.00000000
35	30+	2-4	3	male	0.50000000
36	<10	4+	3	male	0.10810811
37	10-20	4+	3	male	0.21621622
38	20-30	4+	3	male	0.09523810
39	30+	4+	3	male	0.23809524

Tabla 7. Distribución de sobrevivientes por sexo, edad (niños pequeños 2-4 años), clase y tarifa.

El algoritmo de arboles de decisión (yendo un paso más y entrando a machine learning), comienza con todos los datos en el nodo raíz y escanea todas las variables para dividirla mejor, más de lo que pudimos ver en los análisis anteriores. Podemos mirar SibSp, Parch o Embarked. Las variables restantes de nombre del pasajero, número de boleto y número de cabina son identificadores únicos, por ahora, pasamos a construir un árbol de todo lo demás (con rpart de R[2]). En la Figura 1, las decisiones que se han encontrado van mucho más allá de lo que vimos la última vez cuando las buscamos manualmente. Se han encontrado decisiones para la variable SipSp, así como el puerto de embarque que ni siquiera miramos. Y en el lado masculino, los niños menores de 6 años tienen una mejor oportunidad de supervivencia, aunque no haya demasiados a bordo (Stephens, 2014).

2 https://www.r-project.org/

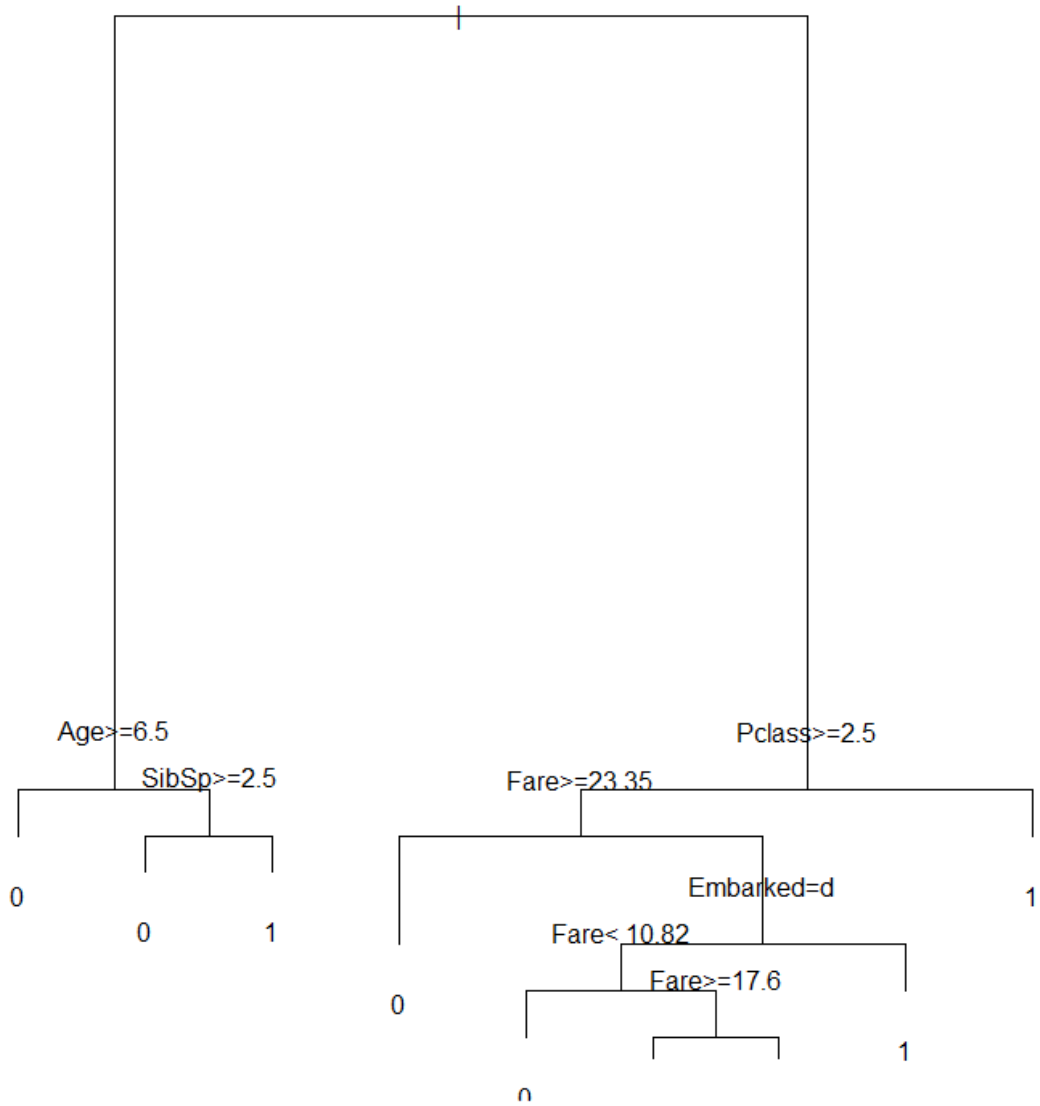

Figura 1. Árbol de decisión para variable Sex, SibSp, Parch y Embarked.

El paquete rpart automáticamente capa la profundidad del árbol, usando una métrica de complejidad que impide que el modelo resultante pierda el control. Cabe destacar que el overfitting hace que un árbol de decisión crezca demasiado, abarcando reglas extensivamente complejas que no pueden generalizar a pasajeros desconocidos (tal vez esa mujer de 34 años de tercera clase que pagó 20,17 dólares por un billete de Southampton con una hermana y madre a bordo puede haber sido un poco de un caso raro), a veces uno puede encontrarse haciendo reglas a partir del ruido que nos ha confundido (Stephens, 2014).

La ingeniería de funciones o de características es importante para el rendimiento del modelo, que incluso un modelo simple con grandes características, puede superar a un algoritmo complejo; es el factor más importante para determinar el éxito o el fracaso de un modelo predictivo: reduce realmente al elemento humano en el aprendizaje automático (la intuición humana y creatividad, marca la diferencia) (Stephens, 2014).

Supongo que es lo que pasa con la calificación lograda jugando con las edades (prediciendo que solo los niños de 2 a 4 años sobreviven), que hasta ahora no se logro mejorar con todo lo ya hecho.

Para aplicar ingeniería de características, podemos empezar por los tres campos de texto que nunca enviamos a los árboles de decisión: número del boleto, la cabina, y el nombre; tal vez partes de esas cadenas de texto podrían extraerse para construir un nuevo atributo predictivo. Los nombres en el barco representaban una sola persona, pero el título de las mismas podría darnos un poco más de penetración (ver Tabla 8), ya que algunos de estos significaban nobleza, edad, etc (Stephens, 2014).

Se debe crear una nueva variable de título, necesitaremos realizar las mismas acciones tanto en el conjunto de entrenamiento como en el conjunto de pruebas, de modo que las características estén disponibles para el crecimiento de nuestros árboles de decisión y para hacer predicciones sobre los datos de prueba no vistos. Una manera fácil de realizar los mismos procesos en ambos conjuntos de datos al mismo tiempo es fusionarlos (agregando la columna Survived para ambos conjuntos, llenando con valores perdidos o Nas en test). Ahora obtenemos 1309 filas con 12 columnas (Stephens, 2014).

Capt	Col	Don	Dona	Dr	Jonkheer
1	4	1	1	8	1
		Lady	Major	Master	Miss
		1	2	61	260
Mlle	Mme	Mr	Mrs	Ms	Rev
2	1	757	197	2	8
				Sir	the Countess
				1	1

Tabla 8. Títulos de las personas a bordo.

En la Tabla 8, hay algunos títulos muy raros, vamos a combinar algunos de los más inusuales pero similares, como Mademoiselle y Madame; Capitán, Don, Mayor y Señor; Dona, Lady, Jonkheer y la Condesa; eran gente rica, y pudieron haber actuado de manera similar debido a su noble nacimiento: combinando estos dos grupos se reduce el número de niveles de factor a algo que un árbol de decisión podría manejar mejor (Stephens, 2014).

Existe otra hipótesis, que si una persona tenía 5 miembros de la familia o más (incluyendo ellos mismos), es más probable que no haya sobrevivido, especialmente en la tercera clase que tenía familias enteras a bordo. Eso supone que la Edad, Sexo y Clase, son los mejores predictores para la supervivencia (ver Figura 2), aunque clase, sexo y tamaño de familia es más contundente (ver Figura 3) (Jason, 2016).

Sería ideal crear una variable de tamaño de familia, ajustada para personas compartiendo camarotes pero no registradas como miembros de una familia. Utilizando Sexo y Edad, la exactitud en entrenamiento ronda el 84,5%, mejor que agregar clase o tarifa, o ambas. Agregar SibSp y Parch, es un poco mejor que las anteriores, pero aún menor que la edad y el sexo. Sin embargo, agregando la variable de familia, mejora a 87.2%. En la Figura 4 se muestra claramente la supervivencia contra la disminución en el tamaño de la familia de 5 o más. Una familia pequeña de 2 a 4 personas aumenta la probabilidad de supervivencia, pero 5 o más, no. Tanto las mujeres como los hombres de familias numerosas aparentemente se quedaron juntos y perecieron juntos, porque no se podían salvar a todos. A veces la adición de todas o muchas variables como factores al modelo no siempre es una buena idea. Algunas veces empeora la exactitud de la predicción, en otras las mejora. Es mejor afinar el proceso de aprendizaje utilizando los parámetros correctos y para el Titanic vemos la importancia de ciertas variables (Survived, Age, Sex, Pclass, *FamilySize*) (Jason, 2016).

Las variables SibSb y Parch, indican el número de miembros de la familia con los que viaja el pasajero, según lo anterior, parece razonable suponer que una familia grande podría tener problemas para localizarse mientras todos se apresuran a bajar del barco que se hunde, por ello agregamos estas dos en una nueva variable, FamilySize. Yendo un poco más allá, podríamos tratar de extraer el apellido de los pasajeros y agruparlos para encontrar familias, pero hay que tener cuidado con los apellidos comunes (como Johnson). Por ello, combinar el apellido con el tamaño de la familia es más viable, ninguna familia debe tener la misma variable de FamilySize en el Titanic, agregamos una nueva columna llamada FamilyID. Siguiendo la hipótesis de que las familias grandes podrían tener problemas para mantenerse juntas en el pánico, eliminemos cualquier tamaño familiar de valor dos o menos, agrupando y renombrado a *small* todas ellas. Es necesario, tener especial cuidado con las variables de tipo factor en R. Ahora estamos listos para dividir los conjuntos de prueba y entrenamiento en sus estados originales (418 y 819), llevando consigo nuestras nuevas y sofisticadas variables de ingeniería (Stephens, 2014).

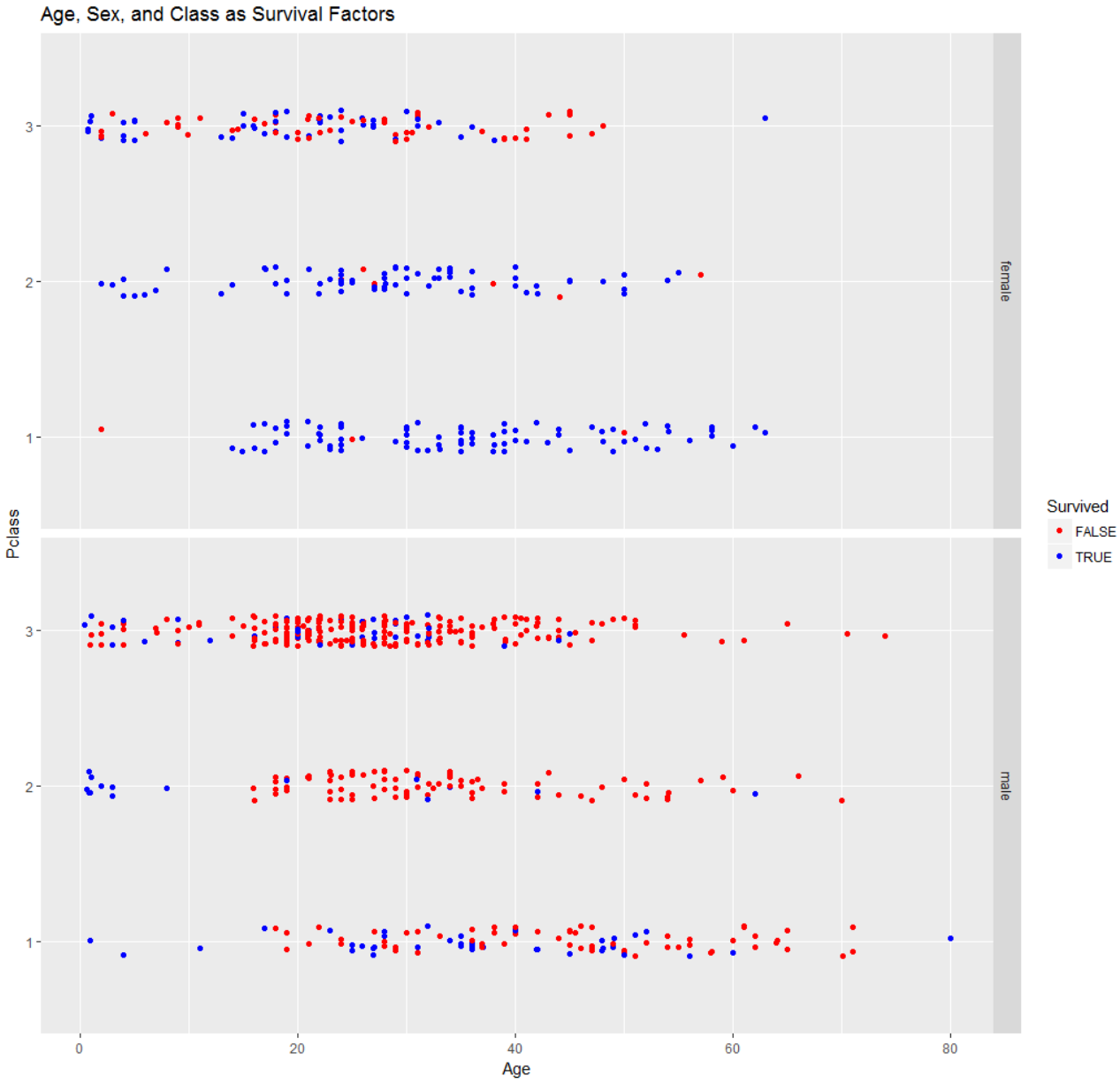

Figura 2. Clase, sexo y edad como factores de supervivencia.

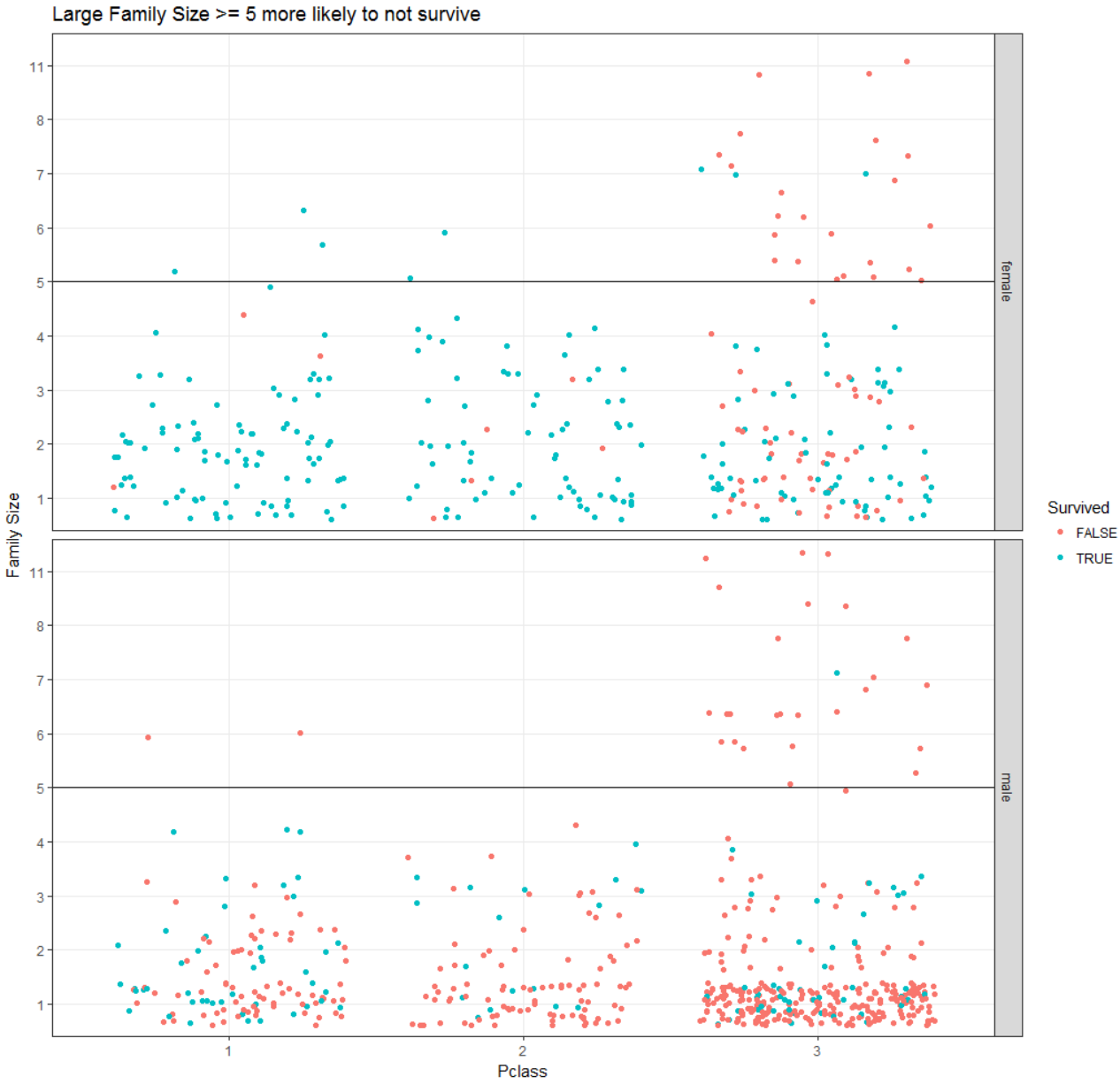

Figura 3. Tamaño de familia, sexo, clase como factores de supervivencia.

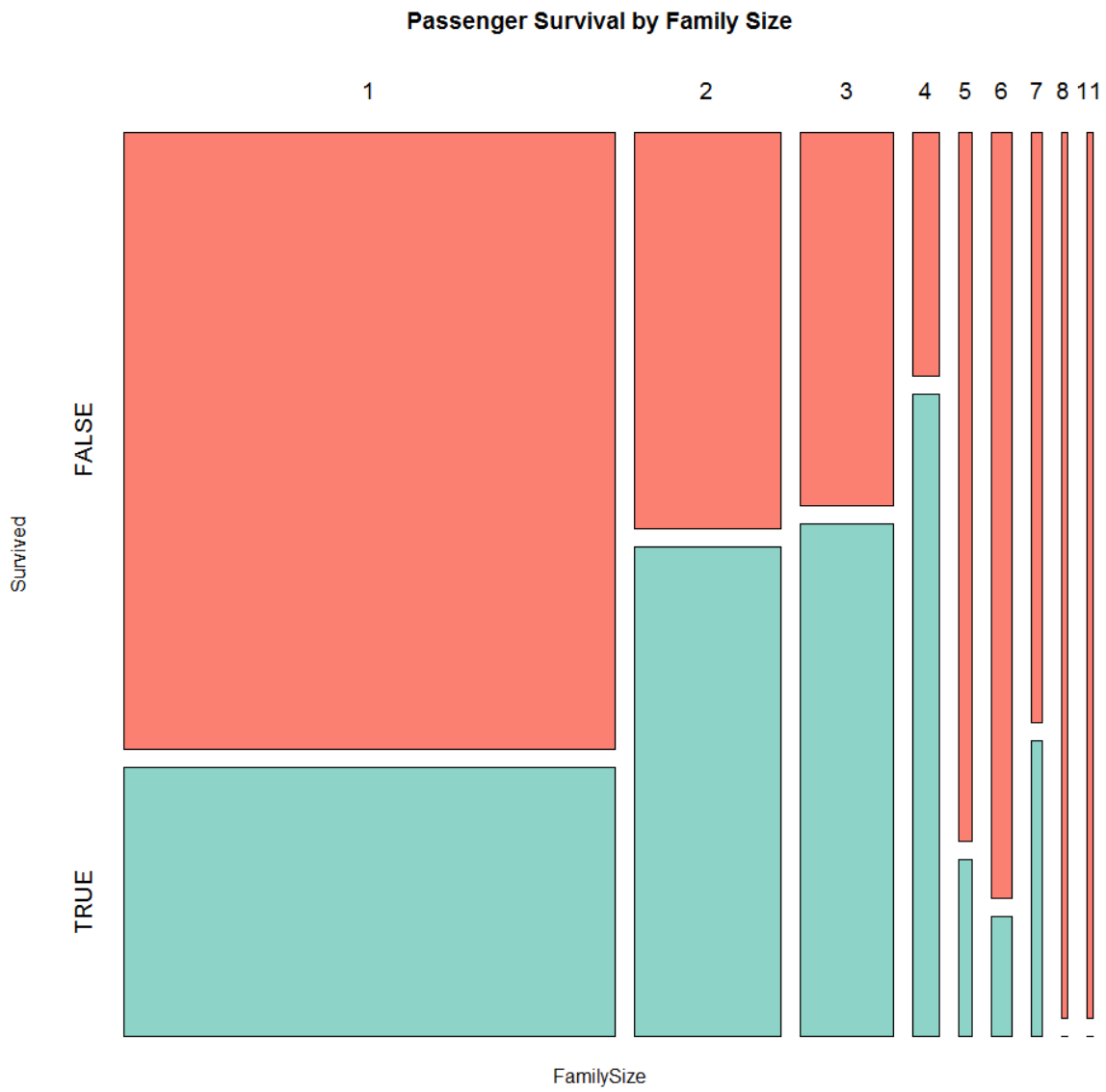

Figura 4. Pasajeros sobrevivientes por tamaño de familia.

El árbol con las nuevas variables se ve en la Figura 5, aunque no tan bien pero sirve para ver la profundidad, las nuevas variables gobiernan básicamente el árbol, puesto que los árboles favorecen las que poseen muchos niveles, como FamilyID. Sin embargo, en la mayoría de los casos, las variables de título o de género regirán la primera decisión debido a la naturaleza codiciosa de los árboles de decisión (Stephens, 2014).

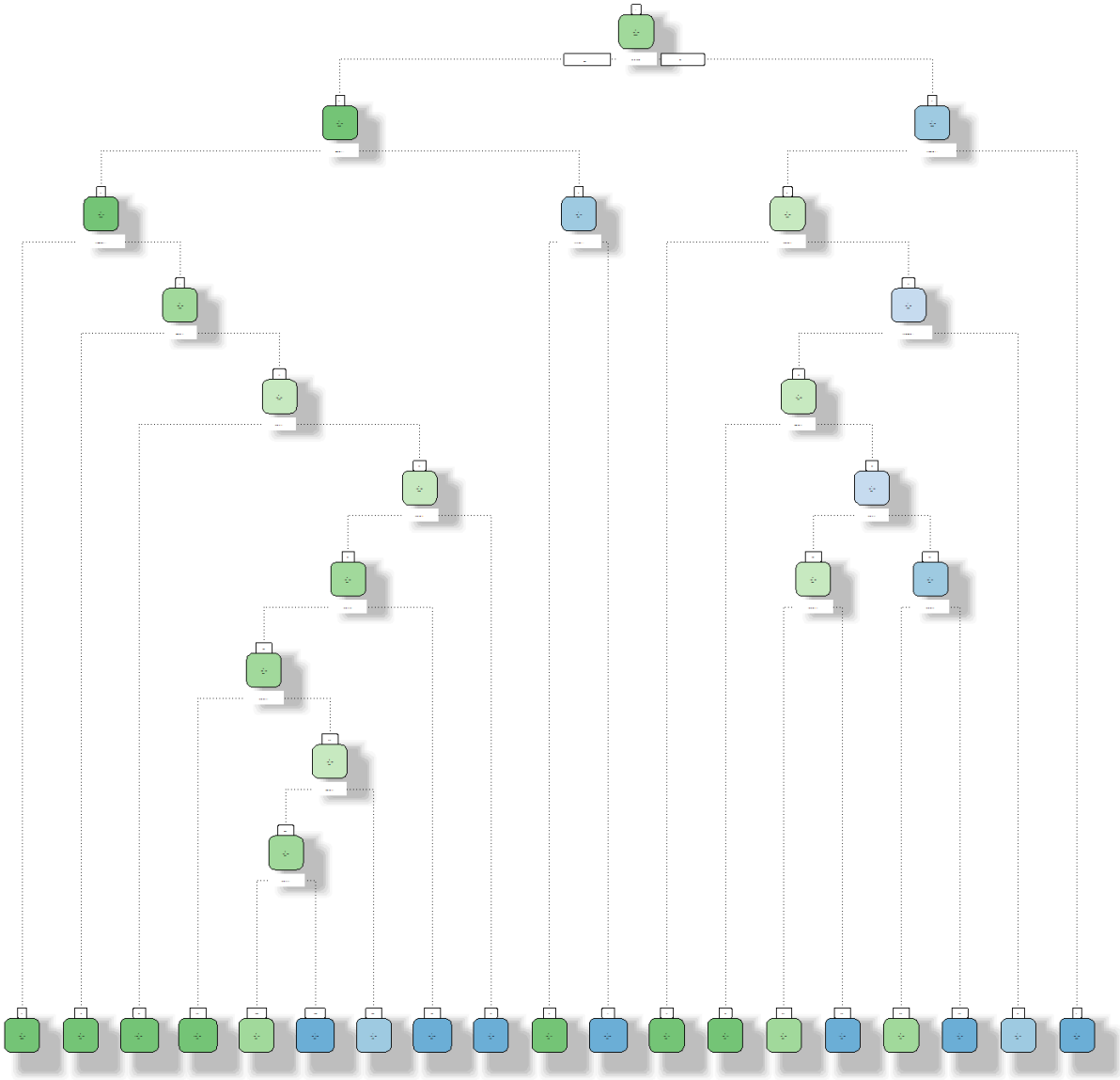

Figura 5. Árbol resultante de las variables obtenidas con la ingeniería de características.

Con estas modificaciones alcanzamos el valor de nuestra predicción anterior, sobre que solamente los niños de 2 a 4 años sobreviven.

Construyendo un conjunto muy pequeño de tres árboles de decisión simples podemos ilustrar los problemas de sobrecarga que presenta este modelo (ver Figura 6), y ver como cada uno de estos árboles toma sus decisiones de clasificación basadas en diferentes variables.

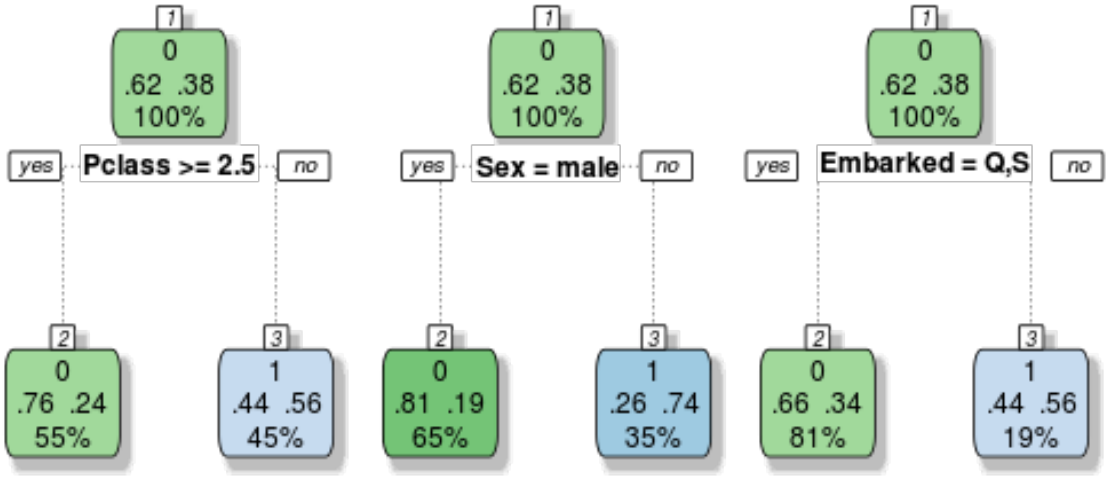

Figura 6. Sobrecarga en árboles de decisión[3].

Los modelos random forest supeditan árboles mucho más profundos que la Figura 3, por defecto crecen hacia fuera en la medida de lo posible, pero como las fórmulas para construir un solo árbol de decisión son las mismas cada vez, se requiere alguna fuente de aleatoriedad para hacer que estos árboles sean diferentes entre sí. Es importante destacar, que si tenemos características muy fuertes como el sexo en el Titanic, dominará la primera decisión en la mayoría de los árboles; sin embargo, la segunda fuente de aleatoriedad en lugar de examinar todo el conjunto de variables disponibles, tomará sólo un subconjunto de ellos, normalmente la raíz cuadrada del número disponible, de esta manera, muchos de los árboles ni siquiera tendrán la variable de sexo disponible en la primera división, y tal vez ni siquiera la vean hasta varios nodos de profundidad. (Stephens, 2014).

En el conjunto de datos hay una gran cantidad de valores de edad que faltan, random forest necesita de estos valores, a diferencia de los arboles de decisión que pueden trabajar sin ellos. Antes habíamos asumido que todos los valores faltantes eran la media o la mediana de los datos restantes (como se pudo ver en la Tabla 3). Hay 263 valores perdidos de 1309 (un 20% de los datos), así que crear un árbol en el subconjunto de los datos con los valores de edad disponibles, y luego reemplazar los que faltan, será conveniente. Además de la edad, a Embarked y Fare le faltan valores de dos maneras diferentes: Embarked tiene un espacio en blanco para dos pasajeros (vamos a reemplazar esos dos con "S" de Southampton, ya que la mayoría embarco allí). Mientras que Fare, tiene un pasajero con una NA, así que hay reemplazarlo con la tarifa mediana. Otra restricción de random forest en R, es que sólo pueden procesar categorías con hasta 32 niveles y FamilyID tiene casi el doble, para reducirlo, podemos cambiar estos niveles a sus enteros subyacentes para tratarlos como variables continuas, o hacerlo manualmente: para ello crearemos una nueva variable, a partir de FamilyID, aumentando el corte para ser una familia "pequeña" de 2 a 3 personas, quedando en 22 niveles (Stephens, 2014).

Una buena práctica para random forest o cualquier otro algoritmo que exija aleatoriedad, es establecer la semilla (el valor no es importante) antes de comenzar para lograr que los resultados sean reproducibles la próxima vez que lo ejecutemos, de lo contrario puede obtener diferentes clasificaciones para cada ejecución.

En R, para random forest, en lugar de especificar method = "class" como con rpart, forzamos al modelo a predecir nuestra clasificación cambiando temporalmente nuestra variable objetivo a un factor con sólo dos niveles usando as.factor(). El argumento importancia = TRUE nos permite inspeccionar la importancia variable y el argumento ntree especifica cuántos árboles crecerán. Es imprescindible acotar que con el bagging aproximadamente el 37% de la filas se quedaría fuera, random forest no desperdicia esas observaciones “out-of-bag” (OOB), sino que las usa para ver cuán bien funciona cada árbol en datos no vistos (un conjunto de pruebas de bonificación para

3 http://trevorstephens.com/kaggle-titanic-tutorial/r-part-5-random-forests/

determinar el rendimiento del modelo sobre la marcha) (Stephens, 2014).

Hay dos tipos de medidas de importancia mostradas en la Figura 7. La exactitud de una prueba para ver cómo de peor el modelo se realiza sin cada variable, y Gini, que esencialmente mide cómo los nodos son puros al final del árbol. La variable de título está en la parte superior para ambas medidas y las variables restantes diseñadas tienen también una importancia alta, hemos acertado al trabajarlas.

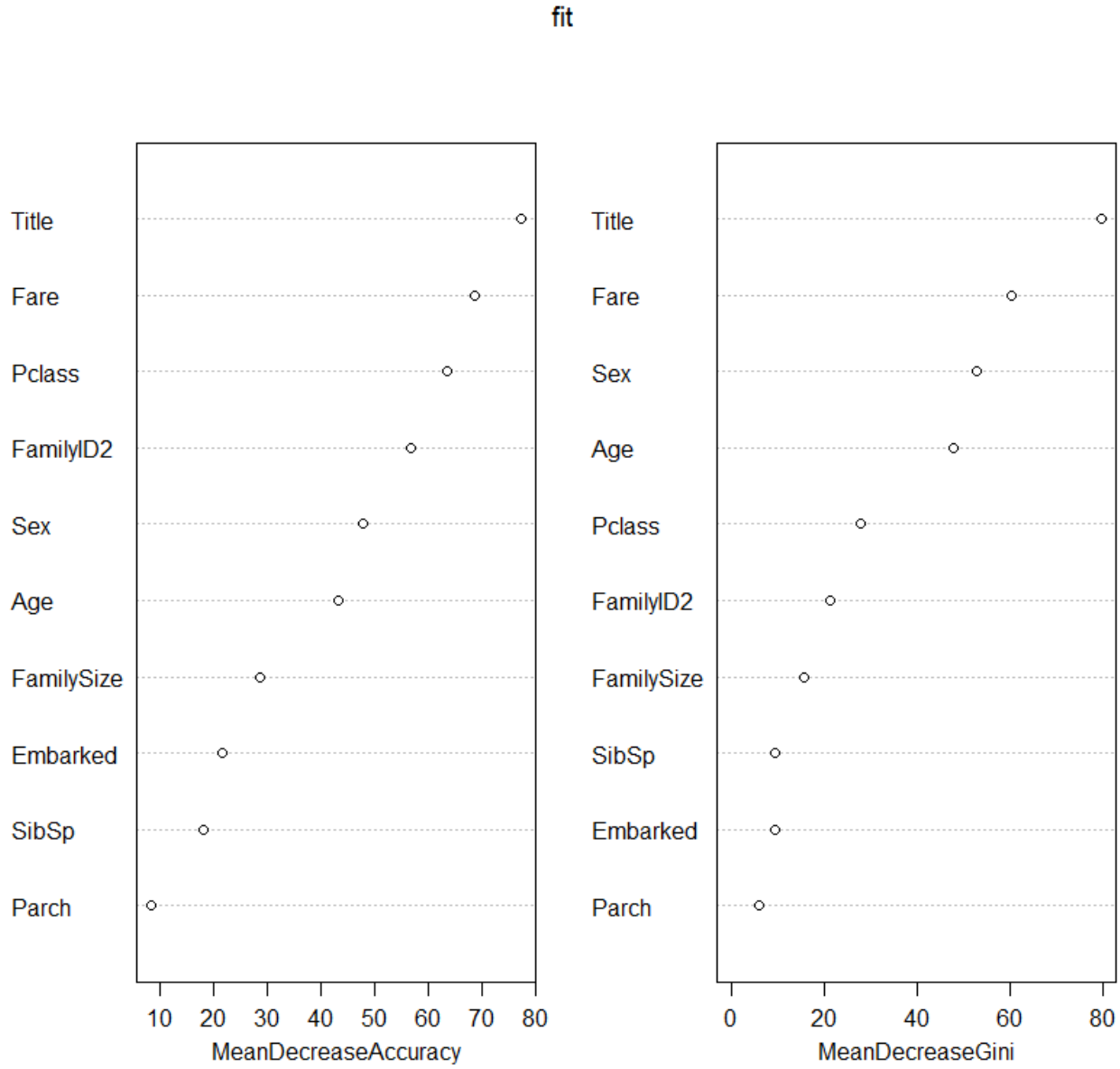

Figura 7. Valores de fit de las variables del conjunto de datos[4].

Una variente de random forest es el que utiliza la inferencia condicional: toman sus decisiones de maneras ligeramente diferentes, usando una prueba estadística en lugar de una medida de bondad, pero la construcción básica de cada árbol es bastante similar (se utiliza el paquete party de R). Estos árboles son capaces de manejar factores con más niveles que los random forest, así que la variable FamilyID se podrá utilizar, en este tipo de árboles se debe especificar el número de árboles dentro de un comando más complicado, ya que los argumentos se pasan de manera diferente y el número de variables a muestrear en cada nodo se pasa manualmente: aplicando este tipo de arbol se logra un acierto de 81,340% en Kaggle (Stephens, 2014).

4 http://trevorstephens.com/kaggle-titanic-tutorial/r-part-5-random-forests/

Volviendo a la ingeniería de características, para cabina, podemos extraer la primera letra como una etiqueta (los números de la cabina corresponden a una clase de pasajeros diferentes, que es un factor crucial para determinar si uno podría sobrevivir). Sin embargo, en esta variable los registros están desaparecidos en su mayoría, por ello pierde su importancia. Para boleto, se puede extraer el primer carácter alfabético para etiquetar el ticket (Xiang, 2016).

La regresión logística en R, se puede aplicar con el paquete glmnet, y la función ridge trabaja un poco mejor que lasso para el conjunto de datos. Se puede incrementar el porcentaje de acierto en Kaggle a 82,297% colocando el valor de la semilla a 1, utilizando como métrica "class", aunque el resultado de esta métrica no es tan estable como con desviación (Xiang, 2016).

Una Máquina de elevación del gradiente o Gradient boosting machine, la podemos aplicar en R, el paquete Caret sirve para ajustarla a través de cuatro parámetros que se pueden sintonizar: n.trees, el número de árboles; interaction.depth, el número de interacción; shrinkage, tasa de aprendizaje, el impacto de cada árbol en el modelo final; y n.minobsinnode, número mínimo de observaciones en los nodos terminales de los árboles. El puntaje enviado a Kaggle es de 81,340%, el tamaño del conjunto de datos es pequeño, y el resultado de GBM no es muy estable para diferentes semillas al azar (Xiang, 2016).

Puesto que tenemos tres modelos con un porcentaje de acierto bueno en Kaggle, podemos utilizar un Ensemble, una forma eficaz de combinar los modelos para reducir la tasa de error global.

Ensemble se encarga de extraer el patrón descubierto por cada modelo para poder lograr una mejor puntuación. Por lo tanto, los resultados no correlacionados de cada modelo puede trabajar mejor, ya que cada modelo contiene patrones diferentes (Xiang, 2016). El resultado del envío a Kaggle es de 82,297%, idéntico al de la regresión logística, pero menos estable.

Técnicas de clasificación

Cuando hacemos el filtrado a mano, para encontrar más subgrupos, de manera fina, con capacidad de predicción, se necesitaría mucho tiempo para ajustar tamaños y mirar la interacción de muchas variables diferentes.

El siguiente paso (y una aproximación a machine learning) son los árboles de decisión: un algoritmo simple y elegante que puede hacer este trabajo por nosotros. Los árboles de decisión tienen una serie de ventajas, son un modelo de glass-box, después de que el modelo haya encontrado los patrones en los datos, se puede ver exactamente qué decisiones se tomarán para los datos ocultos que se desean predecir; son intuitivos, son la base para algunos de los algoritmos de machine learning más poderosos y populares (Stephens, 2014).

Como dice (Stephens, 2014) los árboles de decisión son codiciosos, puesto que hay un gran número de decisiones que podrían hacerse, y la exploración de cada posible versión de un árbol es computacional y extremadamente caro. Aunque por lo general hacen un gran trabajo dada su velocidad, la solución óptima no está garantizada (son propensos a overfitting).

La regresión logística, al igual que random forest o GBM, trabajan mejor en general para este conjunto de datos, obtuve las mejores predicciones aplicándolos. Por otro lado, si hacemos un promedio de los resultados de todos los modelos empleados con un porcentaje de acierto estable (random forest con inferencia condicional, regresión logística y GBM), a veces podemos encontrar un modelo superior, resultante de su combinación, que de cualquiera de sus partes individuales, ya que estos resultados se promedian (o son votados) en todo el grupo. Es por ello que de todas la técnicas, me quedo con el Ensambling[5].

5 https://mlwave.com/kaggle-ensembling-guide/

Presentación y discusión de resultados

En una competencia de Kaggle, donde la clasificación privada aporta sólo el 50% de los datos de prueba a evaluar para nuestras puntuaciones públicas, a veces un modelo más sofisticado no supera a uno simple. Como se dio en el caso del modelo en que niños entre 2 y 4 años sobrevivan (niñas no), luego todos los hombres mueran y las mujeres no: llegué a un 79,4% en Kaggle superando a los árboles de decisión.

Con los árboles de decisión, pese a darle un poco de vueltas al modelo sobre el conjunto de datos, no lograba mejorar mis envíos (inclusive los empeoraba). Con random forest, se obtienen buenos resultados, y ya con los tutoriales existentes se logra un muy buen resultando, escarbando mucho en las variables ya. De todas formas la que mejor resultó para este conjunto, es la variante de inferencia condicional.

La regresión logística con solo intercambiar el valor del parámetro type.measure = ‘deviance’ por ‘class’, permite superar incluso a la variante de inferencia condicional en cuanto al porcentaje de acierto en Kaggle, a coste de perder estabilidad, de todas formas, es un modelo que me sorprendió gratamente.

El aumento de gradiente, dado que el conjunto de datos es muy pequeño, el rendimiento no es estable, puesto que dificulta el ajuste de los parámetros, de todos modos se logra el mismo resultado que utilizando inferencia condicional. Sin embargo, algunas combinaciones de parámetros funcionan bien a través de la validación cruzada en el conjunto de entrenamiento, esto según (Xiang, 2016), y que superponen al conjunto de pruebas.

El método de conjunto, ensamble o ensambling, el apilamiento y la mezcla de los modelos logra mejores resultados, que podría ser el siguiente paso para ajustar el modelo.

Conclusiones

Como mi mejor resultado fue de 82,297% en Kaggle, al ser mi primer experiencia en una competencia de este tipo, creo que es muy buen resultado. Al hacer la predicción manualmente, observando el conjunto, acercándome a los datos logré un 79,4%. No realicé ningún envío con arboles de decisión, puesto que no mejoraban los resultados a los que había llegado escarbando entre los datos. Sin embargo, con random forest, al ser un modelo potente, mejoré a 81,340% utilizando inferencia condicional. En varias ocasiones, tanto con random forest como con árboles de decisión, intenté trasladar mis resultados obtenidos con respecto a la edad (79,4%) pero siempre logrando resultados más bajos o iguales a lo sumo.

La regresión logística me ayudó a saltar varias posiciones (llegando a mi mejor ubicación: 202 con 82,297% en el top 4% de Kaggle), tan solo modificando el valor de un parámetro. Con GBM, pese a leer y escuchar que no daba buenos resultados para este conjunto de datos, igualé el resultado obtenido con la inferencia condicional (81,340%).

Al trabajar el conjunto, hacer un ensamble de todos los modelos aplicados anteriormente me llevó a obtener 82,297%, resultado idéntico al obtenido con la regresión logística, pero éste tiende a ser un resultado más estable que aquel, al tomar los valores de todos los conjuntos.

Trabajo Futuro

Se podría seguir intentando mejorar resultados con la regresión logística, a mi juicio, para luego hacer un ensamble de los resultados con otros modelos. Algunas ideas más que sacar de este conjunto de datos, podrían ser centrarse en los números de boleto o de cabina, extraer algunas ideas de ellos para ver si hay más ganancias posibles, al igual que la carta de cabina o poder hincar la historia para concluir sucesos que nos ayuden a construir un modelo más certero.

Listado de soluciones

En total realicé 15 (hasta el 26/04) envíos a Kaggle, una tuvo errores en la plataforma, por lo que sólo se contabilizaron 14, y otras dos no las agregué a esta ficha, puesto que por error tomé otros CSV, no los que deseaba enviar.

No.	1
Preprocesamiento de datos	Individuos sexo femenino, sobreviven todos
Algoritmos/Software	R
% acierto / conjunto ejemplos etiquetados	0.78675
% acierto / conjunto ejemplos NO etiquetados	0.76555
Posición ranking de Kaggle	NA / 13/04 17:52

No.	2
Preprocesamiento de datos	Individuos sexo femenino, sobreviven todos excepto menores a 4 años, sexo masculino menores a 4 años, sobreviven
Algoritmos/Software	R
% acierto / conjunto ejemplos etiquetados	0.79349
% acierto / conjunto ejemplos NO etiquetados	0.77033
Posición ranking de Kaggle	NA / 13/04 18:06

No.	3
Preprocesamiento de datos	Individuos sexo femenino, sobreviven todos excepto entre 2 y 4 años, sexo masculino de 2 a 4 años, sobreviven
Algoritmos/Software	R
% acierto / conjunto ejemplos etiquetados	0.79924
% acierto / conjunto ejemplos NO etiquetados	0.79426
Posición ranking de Kaggle	NA / 13/04 18:43

No.	5
Preprocesamiento de datos	Individuos sexo femenino, sobreviven todos excepto los de 3° clase con tarifa superior a 20
Algoritmos/Software	R
% acierto / conjunto ejemplos etiquetados	0.80808
% acierto / conjunto ejemplos NO etiquetados	0.77990
Posición ranking de Kaggle	1689 / 14/04 00:40

No.	6
Preprocesamiento de datos	Individuos sexo femenino, sobreviven todos excepto: • los de 3° clase con tarifa superior a 20 • entre 2 y 4 años y tarifa superior a 20

	Individuos sexo masculino, sobrevive ninguno excepto: • clase distinta a 3° y edad de 2 a 4 años • clase 3°, edad menores a 2, tarifa menor a 30 • clase 3°, edad entre 2 y 4, tarifa menor a 20
Algoritmos/Software	R
% acierto / conjunto ejemplos etiquetados	0.81369
% acierto / conjunto ejemplos NO etiquetados	0.77990
Posición ranking de Kaggle	1689 / 14/04 01:30

No.	7
Preprocesamiento de datos	Extracción del título social a partir del nombre, cálculo del tamaño de la familia en función de SibSp y Parch, generación de un ID de familias grandes (3 miembros). Edad, un árbol de regresión Anova a partir de Pclass, Sex, SibSp, Parch, Fare, Embarked, Title y FamilySize; datos perdidos de embarque por el puerto más numeroso ("S") y tarifa por la mediana.
Algoritmos/Software	R / RandomForest / Inferencia Condicional
% acierto / conjunto ejemplos etiquetados	0.85634
% acierto / conjunto ejemplos NO etiquetados	0.81340
Posición ranking de Kaggle	425 / 16/04 09:20

No.	8
Preprocesamiento de datos	Igual que el anterior pero con Age + Sex + Pclass + FamilySize
Algoritmos/Software	R / RandomForest / Inferencia Condicional
% acierto / conjunto ejemplos etiquetados	0.85185
% acierto / conjunto ejemplos NO etiquetados	0.76555
Posición ranking de Kaggle	425 / 16/04 13:40

No.	10
Preprocesamiento de datos	Igual que el anterior pero con Age se reemplaza por AgeClass, que categoriza la edad en menores a 2, 2 a 4, 4 a 10, 10 a 20, 20 a 35 y mayores de 35
Algoritmos/Software	R / RandomForest / Inferencia Condicional
% acierto / conjunto ejemplos etiquetados	0.84175
% acierto / conjunto ejemplos NO etiquetados	0.76555
Posición ranking de Kaggle	425 / 16/04 17:50

No.	11
Preprocesamiento de datos	Idéntico al envío 7
Algoritmos/Software	R / Regresión Logística / Ridge (type.measure = 'class')

% acierto / conjunto ejemplos etiquetados	0.84175
% acierto / conjunto ejemplos NO etiquetados	0.82297
Posición ranking de Kaggle	202 / 16/04 16:35

No.	12
Preprocesamiento de datos	Idéntico al envío 7 con afinado de n.trees, interaction.depth y Disminución de la tasa de aprendizaje
Algoritmos/Software	R / GBM
% acierto / conjunto ejemplos etiquetados	0.82416
% acierto / conjunto ejemplos NO etiquetados	0.81340
Posición ranking de Kaggle	209 / 26/04 22:17

No.	13
Preprocesamiento de datos	Idéntico al envío 7 (GBM + RF Inferencia Condicional + Ridge, type.measure = 'class')
Algoritmos/Software	R / Ensemble
% acierto / conjunto ejemplos etiquetados	0.85521
% acierto / conjunto ejemplos NO etiquetados	0.81340
Posición ranking de Kaggle	209 / 26/04 23:50

No.	14
Preprocesamiento de datos	Idéntico al envío 7 (GBM + RF Inferencia Condicional + Ridge, type.measure = 'deviance')
Algoritmos/Software	R / Ensemble
% acierto / conjunto ejemplos etiquetados	0.85721
% acierto / conjunto ejemplos NO etiquetados	0.82297
Posición ranking de Kaggle	209 / 26/04 23:59

Bibliografía

Jason. (2016, abril). Large families not good for Survival. Recuperado 16 de abril de 2017, a partir de https://www.kaggle.com/jasonm/titanic/large-families-not-good-for-survival
Stephens, T. (2014, enero 10). Titanic: Getting Started With R. Recuperado 13 de abril de 2017, a partir de http://trevorstephens.com/kaggle-titanic-tutorial/getting-started-with-r/
Xiang, C. (2016, septiembre 6). Titanic: A Tutorial to Achieve 0.82297. Recuperado 16 de abril de 2017, a partir de https://byrony.github.io/titanic-a-tutorial-to-achieve-082297.html#titanic-a-tutorial-to-achieve-082297

Anexos

Se adjuntan los siguientes archivos:

- **Rproject**
  - carpeta del proyecto en R, con todos los scripts utilizados en la memoria
- **img**
  - gráficos obtenidos en R
- **data**
  - archivos CSV enviados a Kaggle, creados a partir de los modelos

También disponible en https://github.com/mmaguero/titanic/