Práctica 1 – Competición en Kaggle sobre Clasificación Binaria

Sistemas Inteligentes para la Gestión en la Empresa

Máster Universitario en Ingeniería Informática

Curso 2015 - 2016

Germán Martínez Maldonado

54097023B

germanm@correo.ugr.es

Equipo Kaggle: Germán Martínez Maldonado

Mejor puntuación: 0.79426

Puesto final: 1327

Fecha y hora de la solución: Thu, 05 May 2016 19:59:44

1 Trabajo realizado

El trabajo a realizar consiste en predecir si un pasajero del Titanic sobrevivirá o no en función de diferentes datos como son un identificador único, la clase en la que viaja, su nombre, su sexo, su edad, el número de hermanos / cónyuges a bordo, el número de padres / hijos a bordo, el número de ticket, la tarifa de su billete, la cabina en la que se aloja y el puerto en el que se embarca.

Para esta tarea voy a utilizar el software RStudio para preprocesar los datos y utilizando algoritmos como random forest en diferentes variantes o AdaBoost. Los conjuntos de datos sobre los que trabajar ya están creados por la plataforma Kaggle, existiendo un conjunto de entrenamiento y otro de test. Por la parte del preprocesamiento, el trabajo consistirá en predecir valores perdidos en la edad y crear nuevas variables que nos permitan realizar una mejor clasificación en nuestra predicción, como puede ser calcular el tamaño total de la familia a bordo de un pasajero o distinguir el título de un pasajero.

Para los diferentes algoritmos usados durante el entrenamiento, deberemos ir modificando los parámetros de ejecución del mismo, teniendo especial atención al número de árboles de clasificación que se generan, ya que un número de árboles muy grande para un conjunto de datos muy pequeño puede dar lugar a un sobreajuste del algoritmo al modelo. También hay que tener en cuenta que si discretizamos demasiado los valores de algunas variables, esto puede ser perjudicial para el algoritmo debido a que este tendrá un menor intervalo de valores que usar para la distinción.

Una vez realizada la predicción, mediante operaciones con factores de clasificación cruzada y la diagonal de la matriz, calculamos el porcentaje de acierto sobre el conjunto de validación y si al subir los resultados a Kaggle, este difiere mucho del resultado obtenido, podemos suponer que el algoritmo que estamos empleando se está sobreajustando a nuestro modelo, lo que le impide aprender como es debido.

2 Resumen de soluciones intentadas

N.º solución	**Descripción de la manipulación de los datos aplicada**	**Resumen de los algoritmos y software empleados**	**% de aciertos sobre conjunto de validación**	**% de acierto en Kaggle**	**Posición / Fecha / Hora**
1	En el conjunto de test inicializamos todos los pasajeros como “no supervivientes”.	No utiliza predicción, si el sexo del pasajero es “female” y su edad es menor de 18, asumimos que el pasajero sobrevive. Software usado: R.	-	0.73206	19 Apr 2016 23:41:57
2	-	Define un modelo en el que se busca predecir la supervivencia de los pasajeros basándose en su sexo y su edad. Utiliza el algoritmo de particionamiento recursivo y árboles de regresión “rpart” (method=“class”) de la biblioteca “rpart” de R.	0.7957351	0.75598	20 Apr 2016 00:07:48
3	Elimina del conjunto de entrenamiento todos los elementos de los que no se tienen todos los datos. (No se vuelve a hacer).	Usa el mismo modelo de la solución anterior. Utiliza el algoritmo Random Forest “randomForest” (ntree=5000) de la biblioteca “randomForest” de R.	0.7871148	NA (Error al generar el archivo de resultados)	20 Apr 2016 00:25:52
4	Predice las edades de los pasajeros que faltan mediante “rpart” (method="anova").	Utiliza el mismo modelo y mismo parámetros para “randomForest” en R.	0.7923681	0.76555	21 Apr 2016 21:07:10
5	Igual que el anterior.	Añade la variable “Pclass” al modelo y vuelve a usar el algoritmo Random Forest con los mismos parámetros.	0.8193042	0.76077	21 Apr 2016 23:26:54
6	Añade una nueva variable “TamFamilia” resultante de sumar variables “SibSp” y “Parch” más 1 (el propio pasajero).	Cambia la variable “Pclass” por “TamFamilia” en el modelo. Usa el Random Forest con los mismos parámetros.	0.8148148	0.77033	27 Apr 2016 20:01:25

7	Cambia algoritmo de predicción de edades por “gbm” (n.trees = 5000).	Usa el mismo modelo, pero usando el algoritmo AdaBoost del paquete “gbm”: • method = "gbm" • distribution = "adaboost" • interaction.depth = 1 • n.trees = 5000 • shrinkage = 0.01 • n.minobsinnode = 1	0.811	0.77033	27 Apr 2016 21:38:22
8	Igual que el anterior pero con “n.trees = 10000”.	Añade al modelo anterior las variables “SibSp”, “Parch”, “Fare”, “Embarked” y vuelve a usar el algoritmo AdaBoost para la predicción de supervivientes, pero ahora con “n.trees = 10000”.	0.7969	0.75598	27 Apr 2016 21:50:35
9	Igual que el anterior.	Utiliza solo las variables “Sex”, “Age”, “Fare”, “Embarked” y “TamFamilia” para la predicción usando AdaBoost con los mismos parámetros.	0.7994	0.76077	27 Apr 2016 21:53:22
10	Añade una variable “NivelTarifa” que discretiza las tarifas en 4 grupos en función de los cuartiles de la variable “Fare”.	Predice mediante el uso de las variables “Sex”, “Age”, “TamFamilia” y “NivelTarifa” usando AdaBoost con los mismos parámetros.	0.7953	0.77990	27 Apr 2016 22:15:14
11	Igual que el anterior.	Mismo modelo, pero usando Random Forest con “ntree=10000”.	0.8125701	0.77512	28 Apr 2016 16:47:45
12	Añade una variable “NivelEdad” que discretiza las edades en 4 grupos en función de los cuartiles de la variable “Age”.	Añade “NivelEdad” al modelo y vuelve a usar Random Forest con los mismos parámetros.	0.8193042	0.77033	28 Apr 2016 16:59:10

13	Igual que el anterior.	Usa la variables “Survived”, “Sex”, “Pclass”, “Age”, “TamFamilia” y “NivelTarifa” para la predicción usando AdaBoost con los mismos parámetros.	0.7999	0.77512	28 Apr 2016 17:20:03
14	Cambia los grupos de “NivelTarifa” a 3 equivalentes a los 3 tercios de la variable “Fare”.	Igual que el anterior.	0.8029	0.77990	28 Apr 2016 17:31:03
15	Igual que el anterior.	Usa las variables “Survived”, “Sex”, “Pclass”, “Age”, “TamFamilia” y “Fare” para la predicción usando AdaBoost con los mismos parámetros.	0.8042	0.77990	28 Apr 2016 17:39:12
16	Añade una nueva variable “Familia” con el apellido del pasajero, además la edad se predice con el mismo algoritmo, pero un número de árboles menor (n.trees = 5000).	Usa el mismo modelo para añadiendo la variable “Familia” a la predicción con AdaBoost usando “n.trees = 5000”.	0.6891	0.66507	04 May 2016 23:30:29
17	Igual que el anterior.	Igual que el anterior, pero usando “n.trees = 2000” en el algoritmo AdaBoost.	0,7406	0.76555	04 May 2016 23:35:15
18	Igual que el anterior.	Usa las variables “Pclass”, “Sex”, “Age”, “SibSp”, ”Parch”, “Fare”, “TamFamilia” con el algoritmo AdaBoost con los mismos parámetros.	0.81	0.76555	04 May 2016 23:43:46
19	Igual que el anterior.	En el modelo a predecir consideramos la variable “Survived” un factor de forma explícita y usamos las mismas variables que en el modelo anterior para la predicción con el algoritmo AdaBoost y los mismos parámetros.	0.81	0.76555	04 May 2016 23:46:05

20	Igual que el anterior.	Igual que el modelo anterior, pero añade la variable “Embarked” y usa el RandomForest con “ntree=2000”.	0.8327722	0.78469	04 May 2016 23:53:26
21	Modifica la variable “Familia” agrupando las que tengan menos de 3 personas con el mismo apellido con un denominador común.	Igual que el modelo anterior, pero añade la variable “Familia” y usa el Random Forest con “ntree=2000”.	0.8383838	0.77990	05 May 2016 00:19:47
22	Igual que el anterior.	Igual que el anterior, pero usando “cforest” del paquete “party” con “ntree=2000”.	0.8406285	0.77990	05 May 2016 00:28:03
23	A los 2 únicos pasajeros que no tienen indicado el puerto de embarque les asignamos el puerto “S” debido a que es el mayor número de pasajeros tiene.	Usa el mismo modelo que el anterior, pero para predice mediante “randomForest” con “ntree=2000”.	0.8350168	0.77990	05 May 2016 18:39:41
24	Añade una variable “Titulo” con los títulos distintivos del pasajero: “Capt”, “Col”, “Don”, “Dona”...	Igual que el anterior, pero añadiendo al modelo la variable “Titulo” y quitando la variable “Familia”, predice mediante “randomForest” con “ntree=2000”.	0.8338945	0.78947	05 May 2016 18:59:11
25	Agrupa los valores de la variable “Titulo” en “Caballeros” y “Damas”.	Igual que el anterior.	0.8383838	0.78947	05 May 2016 19:28:12
26	Igual que el anterior.	Igual que el anterior, pero añade la variable “Familia”.	0.8361392	0.77990	05 May 2016 19:37:01
27	Modifica el modelo de predicción de las edades faltantes usando ahora las variables “Age”, “Pclass”, ”Sex”, ”SibSp”, “Parch”, “Fare”, “Embarked”, “Titulo”, “TamFamilia” mediante “rpart” con 'method="anova"'.	Igual que la anterior, pero vuelve a quitar la variable familia, y usa “cforest” con “ntree=2000”.	0.8473625	0.79426	05 May 2016 19:59:44

3 Resultado final

Finalmente la solución que ha dado un mejor resultado consiste en predecir las edades desconocidas mediante el algoritmo de partición recursiva y árboles de regresión “rpart” usando las variables correspondientes a la clase, el sexo, el número de hermanos / cónyuges a bordo, el número de padres / hijos a bordo, la tarifa del billete del pasajero, su puerto de embarque, el título del pasajero y el tamaño de su familia a bordo mediante el método “anova”.

El modelo usado para dicha solución consiste en usar la clase, el sexo, la edad, el número de hermanos / cónyuges a bordo, el número de padres / hijos a bordo, la tarifa del billete del pasajero , su puerto de embarque, el título distintivo del pasajero y el número de componentes de su familia a bordo incluyéndose el propio pasajero. Basándose en este modelo, usando “cforest” con un número de 2000 árboles para entrenar nuestro modelo y realizando una predicción basada en ese entrenamiento he obtenido el resultado de que el 84'736% de las predicciones son correctas, porcentaje que ha caído al 79'426% al validar las predicciones en Kaggle.