Trabajo de Predicción

Diego Andérica Richard
Diego.Anderica@alu.uclm.es
Escuela Superior de Informática
Ciudad Real, España

1 INTRODUCCIÓN

Este trabajo se centrará en la realización de un modelo de predicción basado en *Random Forest*. El objetivo principal de esta práctica es, en la medida de lo posible, y con la técnica asignada, construir un modelo que pueda predecir si una búsqueda de hotel resulta en una reserva o no. Esta técnica consiste en disponer de un conjunto de árboles de decisión, cada uno con una profundidad máxima, donde cada rama es una decisión a tomar para establecer una u otra clase. Al final, una vez que todos los árboles han tomado su decisión, se llevará a cabo una votación para decidir la clase final.

Los datos proporcionados corresponden a un ejercicio donde se supone que nos encontramos en una empresa que, actualmente, se encuentra desarrollando un metabuscador de hoteles a nivel global como podrían ser *Kayak*, *HotelScan* o *Trivago*. Además, se cuenta también con un archivo de datos que corresponden con los clicks recibidos desde un metabuscador de hoteles durante el periodo de una semana.

El lenguaje en el que se ha desarrollado este trabajo ha sido Python, utilizando las diferentes librerías de las que dispone para este tipo de problemas, como *Pandas* o *sklearn*.

2 LECTURA Y VISUALIZACIÓN DE LOS DATOS

En primer lugar, con el fin de realizar una exploración inicial de los datos y su organización en el fichero, se ha cargado el archivo CSV con la librería *Pandas* (Figura 1). Además, se ha hecho un recuento de todas las características y elementos totales de los que se compone el *dataset*, obteniendo un total de **8 características** y **158.161 elementos**.

	date	remite_id	checkin	checkout	adults	children	hotel_id	sale
0	2016-04-01	89	2016-04-18	2016-04-26	2	0	255858	0
1	2016-04-01	89	2016-05-27	2016-05-28	2	2	80563	0
2	2016-04-01	89	2016-06-18	2016-06-19	2	0	165762	0
3	2016-04-01	89	2016-06-18	2016-06-19	2	0	165762	0
4	2016-04-01	89	2016-07-13	2016-08-20	2	0	849	0

Figure 1: Estructura del Archivo

Como se puede observar en la Figura 1, contamos con que algunas de las características son fechas en cadenas de texto, mientras que el resto son características de tipo número entero (*int*). El resumen de los tipos de las características se puede ver en la Figura 2.

```
date          object
remite_id      int64
checkin       object
checkout      object
adults         int64
children       int64
hotel_id       int64
sale           int64
dtype: object
```

Figure 2: Tipos de Datos

Por otra parte, atendiendo a los valores dados para la característica *date*, tal y como se especifica en el enunciado, obtenemos las fechas acotadas a la semana en concreto en la que fueron recogidos los datos (Figura 3).

```
Valores diferentes para la característica 'date':
2016-04-04    25001
2016-04-01    24225
2016-04-05    23563
2016-04-03    23552
2016-04-06    22394
2016-04-07    20342
2016-04-02    19084
Name: date, dtype: int64
```

Figure 3: Semana Datos Recogidos

Una vez obtenidas las características de las que se compone el archivo CSV con los valores posibles que puede tomar cada una, se especifica el significado de las mismas:

- ***date*** (objeto). Fecha del click (acotado a una semana, como se ve en la Figura 3).
- ***remite_id*** (int64). Mercado geográfico del portal.
- ***checkin*** (objeto). Fecha de checkin en la búsqueda.
- ***checkout*** (objeto). Fecha de checkout en la búsqueda.
- ***adults*** (int64). Número de adultos en la búsqueda.
- ***children*** (int64). Número de niños en la búsqueda.
- ***hotel_id*** (int64). Hotel en el que se ha hecho click.
- ***sale*** (int64). Si el click resultó en una reserva (1) o no (0).

Por último, antes de graficar los datos, se ha obtenido una tabla con algunos resultados estadísticos que pueden servir de una primera guía acerca de los datos. Aunque podrían carecer de sentido los resultados de algunas de las características como *hotel_id*, otros como los de *adults*, *children* o *sale* pueden arrojar datos interesantes de cara a afrontar la tarea de predicción (Figura 4).

	count	mean	std	min	25%	50%	75%	max
remite_id	158161.0	47.086842	47.328124	27.0	27.0	27.0	27.0	317.0
adults	158161.0	2.016970	0.397708	1.0	2.0	2.0	2.0	5.0
children	158161.0	0.469819	0.786577	0.0	0.0	0.0	1.0	4.0
hotel_id	158161.0	125089.567295	97022.122353	4.0	4676.0	151728.0	192149.0	398894.0
sale	158161.0	0.019543	0.138425	0.0	0.0	0.0	0.0	1.0

Figure 4: Información Estadística Características Cuantitativas

Observando la tabla anterior, de puede extraer que la cantidad media de adultos en las búsquedas es de dos, con una mínima de uno y una máxima de cinco. Durante la semana de recogida de datos, por otra parte, no fueron habituales las búsquedas con niños, pues la media se sitúa en menos de 0.5 y hasta más allá del percentil 75 no se encuentra un resultado diferente a cero. Por último, la cantidad de búsquedas que resultaron en reservas parece que no fue demasiado elevada, puesto que la media se sitúa en un valor menor de 0.02.

3 TRATAMIENTO Y GRAFICADO DE LOS DATOS INICIALES

Puesto que no existen datos nulos ni errores en el *dataset*, no será necesario un tratamiento previo de este. Una vez obtenidos los datos estadísticos de las principales características, se procedió a graficar los valores, de manera que se pudieran observar de un vistazo y de una forma más intuitiva los valores de los que se compone el conjunto de datos.

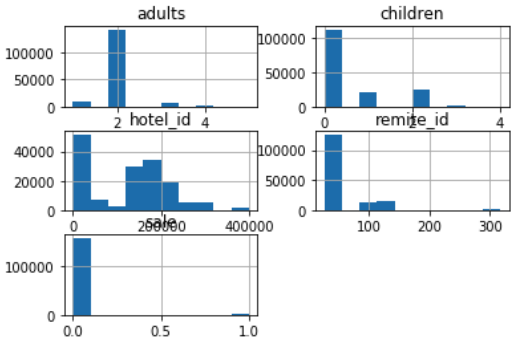

Figure 5: Histogramas Características

4 PREDICCIÓN

Para la realización del trabajo de predicción utilizando la técnica asignada *Random Forest*, se ha hecho uso de la función *RandomForestClassifier*, contenida en la librería *sklearn*. Esta técnica se basa en disponer de un conjunto de árboles de decisión, donde cada uno lleva a cabo una predicción según las reglas que se han construido y de los datos con los que se ha entrenado. Una vez que todos estos árboles han tomado una decisión se lleva a cabo un proceso de votación, donde se extrae una predicción final basada en la predicción mayoritaria de todos los árboles que componen el «bosque». Esta predicción consistirá en una clasificación positiva (1) o negativa (0).

No obstante, antes de llevar a cabo la creación del modelo, se deben extraer conjuntos de datos para entrenar y probar el modelo. Para esta tarea, al tener los datos de una semana completa, se ha decidido separar estos según el día (del 01-04-2016 al 04-04-2016 para entrenamiento y del 05-04-2016 al 07-04-2016 para evaluación). Esto representa prácticamente una separación del 60%/40%. De esta manera, se dispondrán de 91.862 registros para entrenamiento y 66.299 para evaluación.

Una vez se tuvieron los datos separados en dos conjuntos diferentes, se procedió a la definición del modelo. No obstante, esta tarea requiere de una especificación previa de parámetros que se deben fijar al llamar al método *RandomForestClassifier*, como la profundidad máxima de los árboles, la cantidad de estimadores que contendrá el modelo o el criterio de selección que, por defecto, se trata de *gini*, midiendo la pureza del corte (probabilidad de no sacar dos registros con el mismo valor para la variable objetivo dentro del mismo nodo). Para los otros dos parámetros se ha llevado a cabo un estudio en el que se ha probado con diversos valores, tratando de obtener el máximo *score* (precisión media de las predicciones). Además, se ha fijado el parámetro *random_state* a 0 para obtener resultados deterministas entre ejecuciones y *n_jobs* a -1 para aprovechar todos los núcleos/procesadores del ordenador. Las diferentes ejecuciones (Figuras 6, 7 y 8) arrojaron los siguientes resultados:

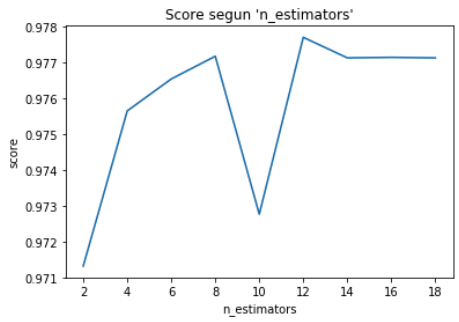

Figure 6: *Score vs n_estimators*

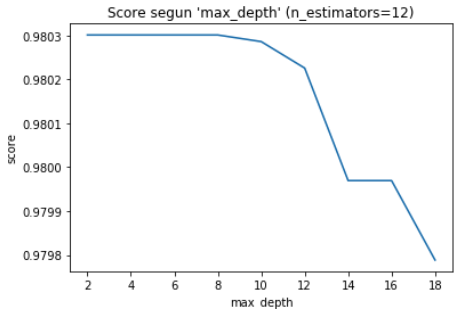

Figure 7: *Score vs max_depth*

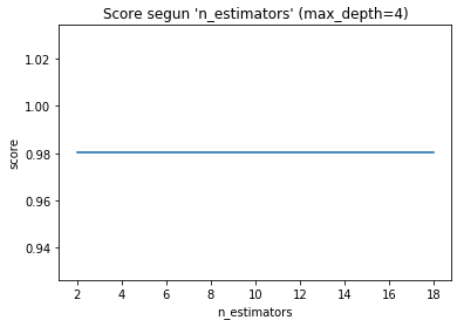

Figure 8: Estudio final

Por tanto, se procedió a la creación del modelo con una profundidad de árbol de 4 y 12 estimadores, lo que arrojaba una precisión del 98.030% y que, a priori, podría parecer bastante aceptable. A continuación, se graficaron las importancias de las características del modelo (Figura 9), observando que la característica *adults* acaparaba la mayor relevancia, seguida por *hotel_id*, *remite_id* y *children*.

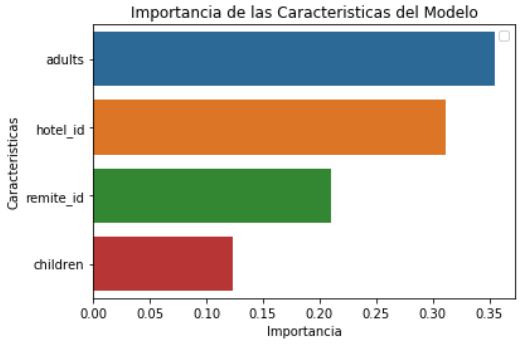

Figure 9: Relevancias de las características

Finalmente, se obtuvo la matriz de confusión para comprobar cuántos fallos y aciertos reales proporcionaba el modelo, observando que predecía muy bien los resultados negativos (0) y fallaba todos los positivos (1) (Figura 10). Principalmente, esto sucedía porque únicamente predecía la clase (0). Por tanto, se decidió llevar a cabo otros dos modelos: uno con balanceo de características y otro con balanceo y extracción de nuevas características, en pos de mejorar la predicción.

```
array([[64993,     0],
       [ 1306,     0]])
```

Figure 10: Ejemplo matriz de confusión modelo 1

Modelo 2: Balanceo

En este modelo se llevó a cabo el mismo estudio de parámetros que el realizado en el primero con la salvedad de que, en este caso, se especificó el parámetro *class_weight* de *RandomForestClassifier* como *balanced*. Esto se debe a que el conjunto de datos inicial, como se ha comentado anteriormente, se encuentra muy descompensado, con menos del 2% de casos positivos (1). De esta forma, según la documentación de la librería, «se usan los valores de y para ajustar pesos de manera inversamente proporcional a la frecuencia de las clases de entrada», compensando así la diferencia entre ambas clases. Los resultados de las ejecuciones se encuentran reflejados en las Figuras 11, 12 y 13.

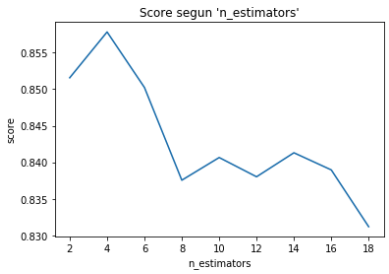

Figure 11: *Score vs n_estimators* (modelo 2)

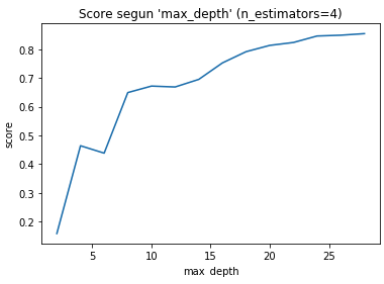

Figure 12: *Score vs max_depth* (modelo 2)

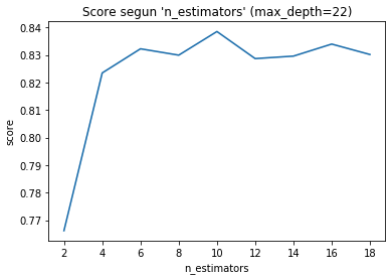

Figure 13: Estudio final (modelo 2)

Finalmente, se construyó el modelo con una profundidad de 22 y 10 estimadores, lo que proporcionaba una precisión del 83.853%. Por otra parte, la importancia de cada una de las características había cambiado drásticamente con respecto al primer modelo, teniendo ahora *hotel_id* el mayor peso (Figura 14).

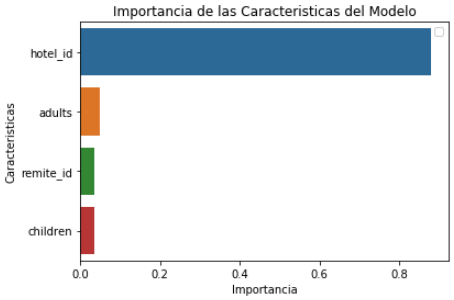

Figure 14: Relevancias de las características (modelo 2)

Por último se realizaron varias pruebas, obteniendo predicciones con diferentes conjuntos de datos para comprobar los aciertos y fallos a través de la matriz de confusión. Un ejemplo una de estas ejecuciones se muestra en la Figura 15. En ella se puede observar que ahora, aunque falla alrededor del 15% de casos negativos (0), consigue acertar una mayor cantidad de casos positivos (1) con alrededor del 50% de precisión.

```
[[52467  9572]
 [  583   643]]
```

Figure 15: Matriz de confusión (modelo 2)

Modelo 3: balanceo y extracción de características

Este tercer modelo se podría considerar una variante del modelo anterior, con la adición de dos características adicionales. En este caso se ha calculado la estación del año de la fecha del *checkin*, así como si la búsqueda se realiza en fin de semana o entre semana atendiendo a las fechas reflejadas en la característica *date*. De esta forma, el estudio de los parámetros arrojó los resultados que se muestran en las Figuras 16, 17 y 18.

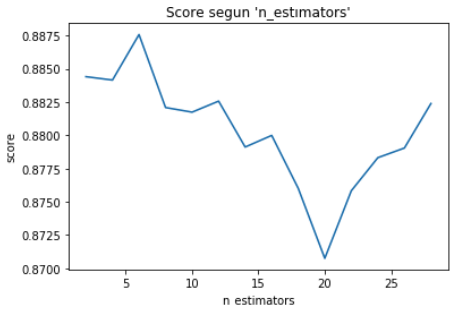

Figure 16: *Score vs n_estimators* (modelo 3)

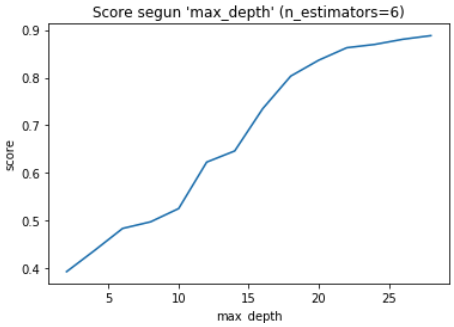

Figure 17: *Score vs max_depth* (modelo 3)

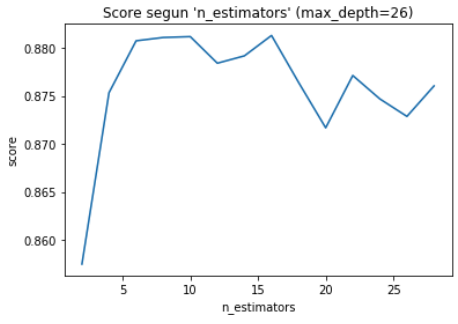

Figure 18: Estudio final (modelo 3)

En este caso, el modelo construido con una profundidad máxima de 26 y 16 estimadores obtenía una precisión del 88.133%. Del mismo modo, se han extraído las diferentes relevancias de las características, donde se puede observar que sucede prácticamente lo mismo que en el modelo anterior, con las nuevas características ocupando las últimas posiciones de relevancia (Figura 19).

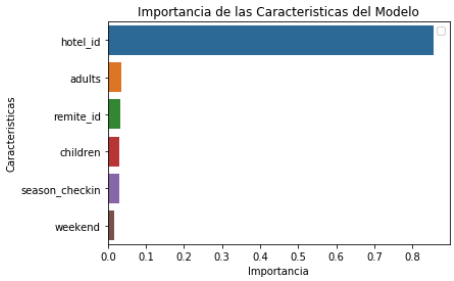

Figure 19: Relevancias de las características (modelo 3)

Finalmente, atendiendo a los resultados de las ejecuciones de las matrices de confusión (en la Figura 20 se tiene un ejemplo de una de ellas), se refleja que la media de los resultados apenas ha variado con respecto al modelo anterior, aunque se ha mejorado la predicción de casos negativos (0). Por tanto, este se considera el mejor modelo de entre los tres que se han construido.

```
[[55101  6945]
 [  617   602]]
```

Figure 20: Matriz de confusión (modelo 3)

5 CONCLUSIONES

Este trabajo ha servido para tratar y aplicar a un caso la técnica de predicción de *Random Forest*. Además, se ha tenido que lidiar con un conjunto de datos poco balanceado, ya que había una gran diferencia entre casos positivos y negativos en cuanto a la clasificación de los elementos. No obstante, se ha tratado de solucionar este «problema», obteniendo predicciones llevando a cabo un estudio de parámetros y un balanceo previo de dichos datos.

REFERENCIAS

[1] **Librería sklearn**. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[2] **Librería Pandas**. http://pandas.pydata.org/pandas-docs/stable/
[3] Material de la asignatura de Desarrollo de Sistemas Inteligentes.