2016/2017

Ingeniería basada en el conocimiento

Opiniones sobre hoteles

Héctor Rodríguez Salgado & Marta Loriente Nieves

Máster en Ingeniería Informática – ESEI – Universidad de Vigo

Contenido

[Introducción 1](#_Toc469327671)

[Exploración de datos 1](#_Toc469327672)

[Análisis de predicción de atributos 2](#_Toc469327673)

[Lazy learning 2](#_Toc469327674)

[Decision trees 3](#_Toc469327675)

[Rules 3](#_Toc469327676)

[Clustering 4](#_Toc469327677)

[K-Means 5](#_Toc469327678)

[EM 6](#_Toc469327679)

# Introducción

El objetivo principal del análisis de datos es la construcción y comparación de diferentes modelos de minería de datos, enfocando la discusión en sus resultados.

Se evaluarán un conjunto de datos siguiendo tres enfoques de análisis diferentes: modelos predictivos, modelos de agrupamiento y modelos de asociación.

Todos estos modelos se construirán utilizando la herramienta de minería de datos WEKA y sus algoritmos tal como se proporcionan, sin modificar ninguno de sus detalles de implementación.

# Exploración de datos

Para realizar los análisis se ha escogido un conjunto de datos de alrededor de 12.000 instancias que recogen las opiniones de los clientes que han visitado distintos hoteles. Dicha información se ha obtenido del buscador de hoteles *Tripadvisor*.

En un inicio, el *dataset* contenía la siguiente información:

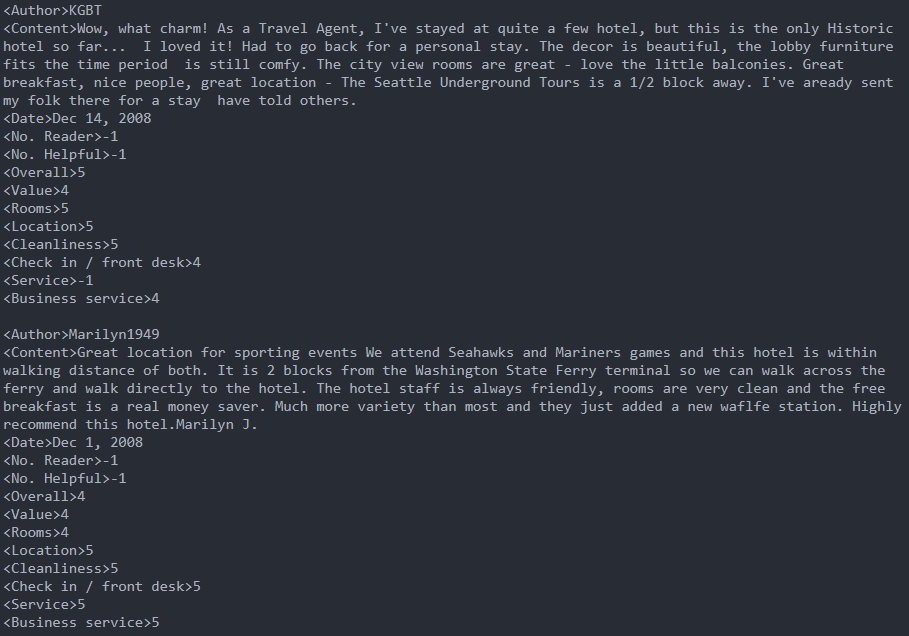


Ilustración - Dataset inicial

Para el análisis, únicamente se requería la opinión del cliente. Por tanto, se ha realizado un preprocesado de los *datasets*, obteniendo como resultado:

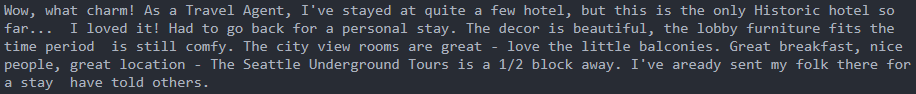


Ilustración - Dataset tras el procesado

De esta forma, cada instancia contiene la opinión de un cliente en un hotel.

Una vez realizado dicho proceso, se han escogido las 50 palabras que más se repiten en todas las opiniones para determinar si la opinión del cliente ha sido buena o mala hacia el hotel en el que se ha hospedado. Dichas palabras pasarán a ser los atributos de interés para el análisis.

A continuación, se muestra el listado de las palabras escogidas: “hotel, small, bed, parking, rooms, friendly, good, got, view, price, seattle, helpful, day, bathroom, area, like, excellent, night, comfortable, desk, didnt, breakfast, dont, little, city, best, recommend, street, hotel, place, floor, free, just, restaurant, staff, right, clean, great, room, nice, really, stay, lobby, stayed, hotels, service, nights, location, time, did, walk”.

# Análisis de predicción de atributos

La primera tarea es determinar si es posible predecir si la opinión de un cliente es buena o mala en base a las palabras que contiene su opinión. Para ello se han ejecutado diferentes algoritmos predictivos y se ha observado el resultado.

## Lazy learning

El primer algoritmo que se ha decidido utilizar ha sido de tipo “lazy learning: (k-) nearest-neighbour”, conocido como IBK en WEKA. Se ha decidido realizar una predicción para cada atributo del dataset, variando el número de vecinos en cada análisis. La tendencia en cada uno de los análisis es que a más vecinos, mejor es la predicción realizada, ya que el porcentaje de acierto es mayor.

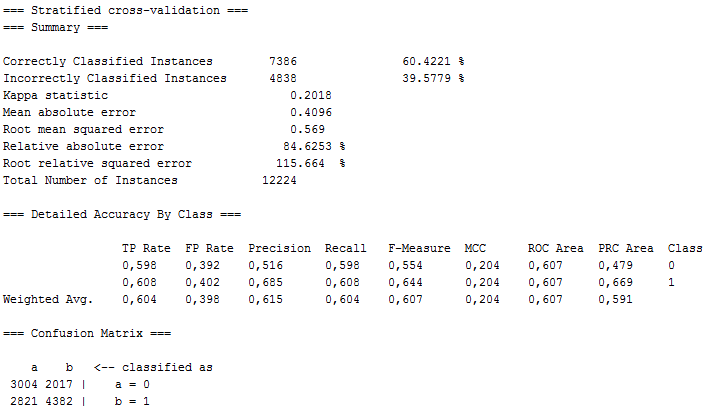


Ilustración - IB1

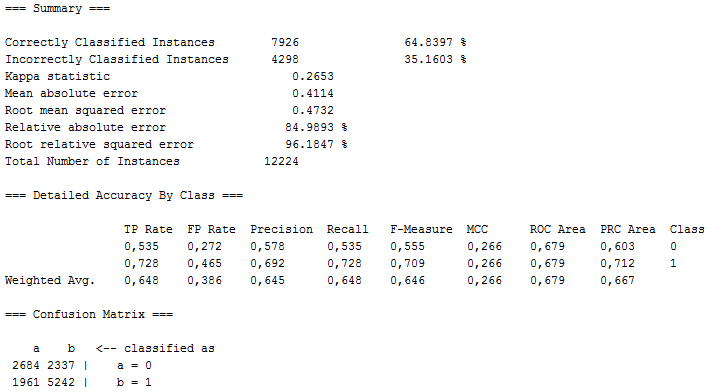


Ilustración - IB5

Como se puede observar en los resultados anteriores, a mayor número de vecinos, mayor es el porcentaje de acierto de este algoritmo. Aunque observando los aciertos obtenidos, se reduce considerablemente el número de instancias clasificadas correctamente para el tipo ‘b’ (valoraciones positivas sobre hoteles) y, en cambio, aumenta para el tipo ‘a’ (valoraciones negativas sobre hoteles). Esto indica que no siempre que se obtenga un porcentaje de acierto mayor, el análisis va a ser mejor, como ocurre en este caso, ya que dicho porcentaje, a efectos prácticos, sólo se refleja en las valoraciones negativas, no en las valoraciones positivas.

## Decision trees

El concepto principal detrás de aprendizaje árbol de decisión es que, a partir de los datos de entrenamiento, se va a construir un modelo predictivo que se asigna a una estructura de árbol. El objetivo es lograr la clasificación perfecta con el número mínimo de la decisión, aunque no siempre es posible debido al ruido o inconsistencias en los datos.

Dentro de este tipo de algoritmos se ha decidido realizar el análisis con el tipo J48.

El resultado que se ha ofrecido utilizando un testeo de cross-validation de 10 folds es el siguiente:

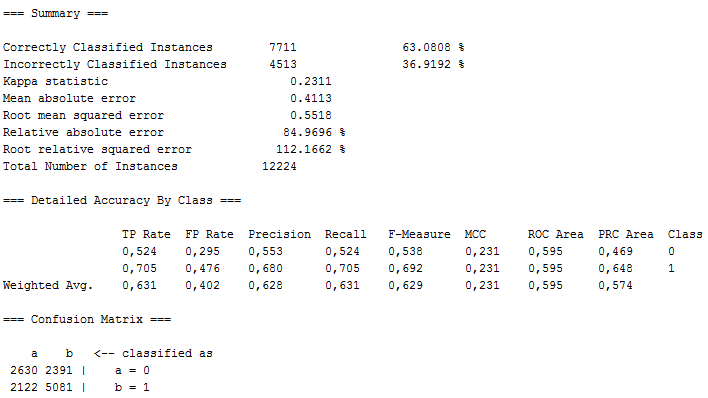


Ilustración - J48

Se toma como atributo clase las valoraciones siendo 0 las valoraciones negativas y 1 las positivas. Tal y como se ve en la ilustración anterior está cometiendo más error cuando clasifica las valoraciones positivas. Con ello se puede decir con certeza que este algoritmo no es bueno para realizar un estudio para este caso.

## Rules

Dentro de este tipo de algoritmos el primero que se va a utilizar será JRIP cuyo objetivo es podar para cortar la reducción de errores, obteniendo como resultado en base al atributo *valuation* como clase del algoritmo:

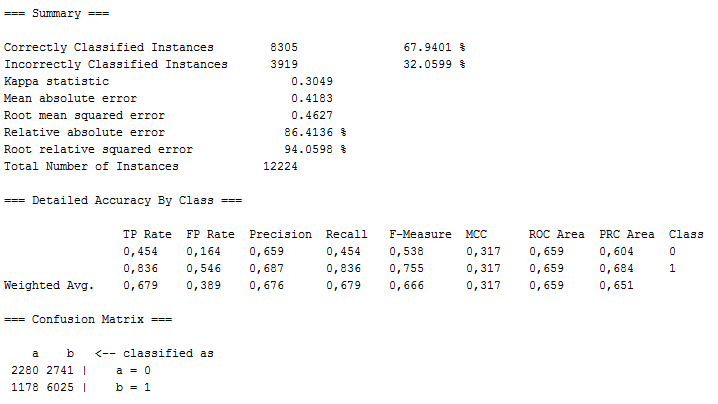


Ilustración - JRIP

Como el resto de algoritmos vistos hasta ahora se puede que se mantiene el tanto por ciento de instancias clasificadas correctamente, pero si se observa detenidamente la matriz de confusión se produce menos errores al predecir las valoraciones positivas como negativas. Por lo tanto, en base a esta última afirmación se puede decir que es hasta el momento el mejor algoritmo de predicción para este caso.

El siguiente algoritmo de este tipo será OneR manteniendo el atributo clase *valuation* obteniendo más o menos el mismo resultado que con el algoritmo JRIP. Sin embargo, utilizando el algoritmo ZeroR los resultados son bastante distintos como se puede ver en la siguiente imagen:

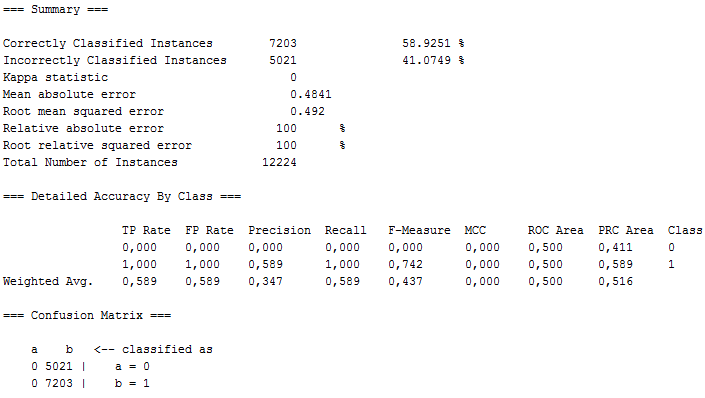


Ilustración : ZeroR

A la hora de predecir las valoraciones negativas no tiene ninguna precisión, con lo cual este algoritmo queda completamente descartado.

A continuación, se muestra el resultado obtenido para el algoritmo PART:

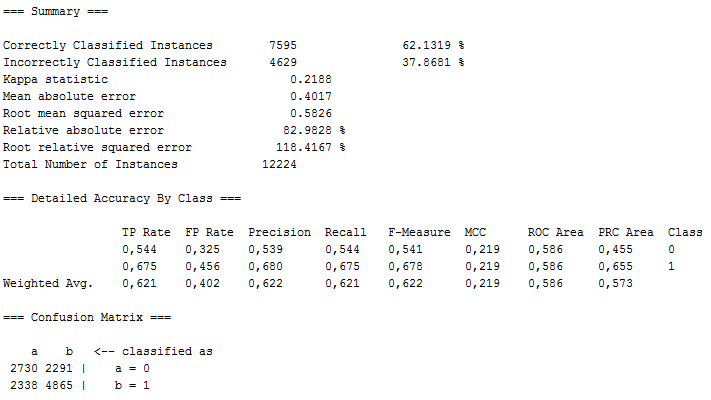


Ilustración - PART

Se mantienen los resultados medios obtenido con los primeros algoritmos aplicados, pero se sigue obteniendo un error al predecir las valoraciones positivas de la mitad de las instancias.

Con respecto a los árboles de decisión se puede afirmar que nos aporta peores resultados, con lo cual queda eliminado.

**Conclusión** para el análisis de atributos es que el mejor algoritmo para este .arff es el algoritmo **rules** porque es el que obtiene mayor porcentaje de instancias clasificadas correctamente y los errores que se comenten en la predicción, tal y como se muestra en la matriz, son los menores. Por lo tanto, tiene mayor precisión a la hora de clasificar los atributos.

# Clustering

El *clustering* o agrupamiento es la tarea de agrupar un conjunto de objetos de tal manera que los miembros del mismo grupo (llamado clúster) sean más similares, en algún sentido u otro. ES la tarea principal de la minería de datos exploratoria y es una técnica común en el análisis de datos estadísticos.

## K-Means

Es un método de agrupamiento, que tiene como objetivo la participación de un conjunto de *n* observaciones en *k* grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano.

Analizamos con el número de clúster igual a 10. También ponemos el atributo displayStdDevs a True (atributo que nos mostrará información más detallada de los clústers).

Vamos a ejecutar el .arff con el modo de testeo *Classes to cluster evaluation* con el atributo *valuation* como clase.

A continuación, se explica paso a paso las partes que se han obtenido al ejecutar este dataset. La primera captura nos muestra el número de iteraciones junto con el error de la suma cuadrática, que se ha comprobado que cuánto mayor número de clúster es menor. La siguiente sección muestra qué puntos aleatorios se pusieron al inicio para saber si son iguales, distintos…

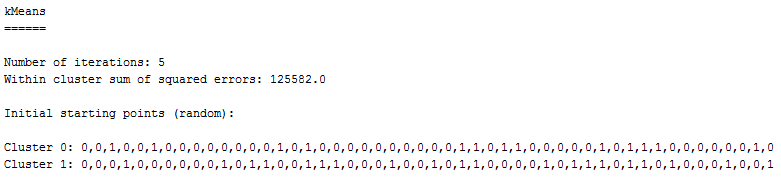


Ilustración - SimpleKMeans

Se muestran ahora los clústers que se han realizado, y la aparición de las palabras en cada uno de ellos.

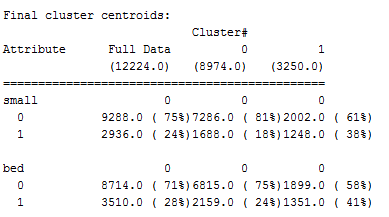
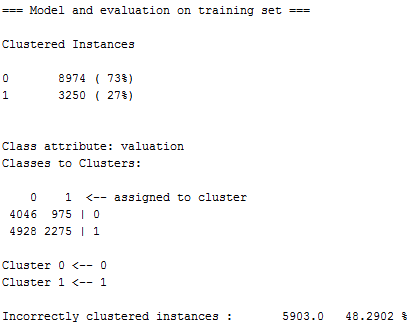


Ilustración - SimpleKMeans

Tal y como se muestra en la ilustración anterior se han realizado 2 agrupaciones, con tantos miembros que se indica justo debajo del número del clúster. Seguidamente se toma cada una de las palabras del dataset y se detalla la aparición en cada clúster (valoraciones negativas y positivas). Este nivel de detalle es gracias al atributo mencionado anteriormente *displayStdDevs*.

Para finalizar se interpretan los clusters obtenidos:

Como se puede ver se ha obtenido el número de instancias para cada uno de los clústers. Es decir, 8974 ocurrencias se han producido en el clúster 0, valoraciones negativas, lo que es un 73% del total. El resto son valoraciones positivas para hoteles.

Ilustración : SimpleKMeans

En la matriz de confusión se ve que se producen más fallos en la clasificación de valoraciones positivas, y el porcentaje de instancias incorrectamente asignadas es casi del 50%

## EM

El algoritmo de expectativa de maximización es un método iterativo para la búsqueda de la máxima verosimilitud, teniendo en cuenta variables latentes.

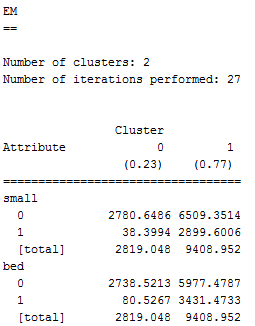


Ilustración : EM

Aplicamos este algoritmo con un número de clúster igual a 2. En este caso ya se pueden ver los resultados. En cada uno de estos clústers se puede observar la cantidad de instancias en cada uno de ellos. En el clúster 0 (valoraciones negativas) tiene 23% de las instancias y en el clúster 1 (valoraciones positivas) tienen un 77%.

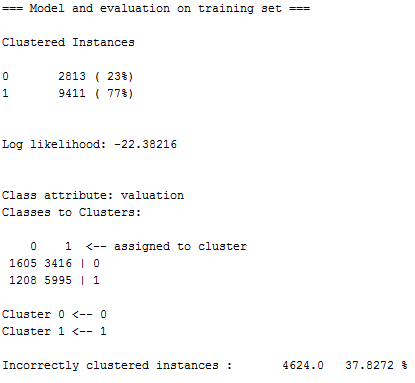


Ilustración : EM

De nuevo se pueden observar ambos clústers y el porcentaje junto al número de instancias asignados a cada uno. Más abajo se observa la matriz de confusión viendo que con respecto a las valoraciones negativas se producen más fallos que aciertos, mientras que con las valoraciones positivas la cantidad de instancias clasificadas erróneamente son menores.

El porcentaje de instancias clasificadas incorrectamente en los clústers es de un 37%.

Conclusión con respecto al clústering es mucho mejor el algoritmo EM debido a que comete menos errores y tiene un menor porcentaje de errores.

## Associate

Este tipo de algoritmo consta de unas reglas de asociación y se utilizan para descubrir hecho que ocurren en común dentro de un determinado conjunto de datos.

En este apartado vamos a tener en cuenta será el algoritmo *A priori*. El cual permite encontrar de forma eficiente “conjuntos de ítems frecuentes”, los cuales sirven de base para generar reglas de asociación. Para comprender este algoritmo es importante primero aclarar ciertos aspectos que se verá en las reglas. Estos aspectos son:

* Soporte (Suport): mide la importancia del conjunto de artículos basándose en el número de transacciones que contienen el conjunto de elementos.
* Confianza (Confidence): mide la fuerza de la regla basándose en la probabilidad condicional de que una transacción que tenga X y también contenga Y.
* Elevación (Lift): mide la influencia de la LHS en la presencia del RHS. Si es igual a uno los conjuntos son independientes, si es mayor los conjuntos están positiva correlacionados, es decir, sabemos el grado en que éstas dos ocurrencias son dependientes entre sí y hace que estas reglas sean potencialmente útiles. Si es menor que uno significa que están negativamente correlacionados, es decir, aparecen con menos frecuencia.
* Convicción (Conviction): puede interpretarse como la frecuencia esperada que X tiene sin Y (es decir, la frecuencia con la que la regla hace una predicción incorrecta), si X e Y fueran independientes, divididos por la frecuencia observada de predicciones incorrectas. Si tiene un valor alto significa que el consecuente depende fuertemente del antecedente.
* El apalancamiento (Leverage): es la proporción de ejemplos adicionales cubiertos tanto por el LHS como por el RHS por encima del esperado si son independientes.

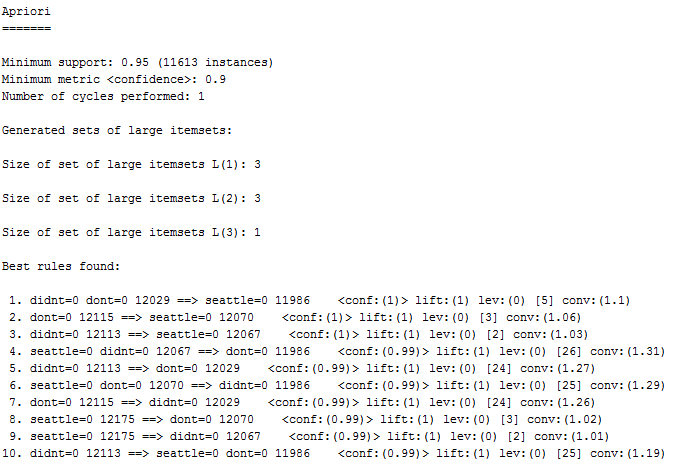


Ilustración : A priori

Como se puede comprobar en la ilustración 14 se han originado las 10 mejores reglas. Estas reglas se han basado en cuatro aspectos: confianza (confidence), elevación (lift), aplacamiento (leverage) y convicción (conviction).

Entonces la mejor regla tiene de confianza 1, lo cual significa que en todas aquellas instancias que haya el atributo didn’t y don’t obligatoriamente tendrán Seattle. Tiene de elevación 1, esto quiere decir que los conjuntos son totalmente independientes.