CAOS uc3m

PRÁCTICA 2 (Parte 1 de 2)
Data Mining

Inteligencia Artificial en las Organizaciones

Grado en Ingeniería Informática

Curso 2024/2025

INTRODUCCIÓN

La minería de texto (text mining) es un área de la inteligencia artificial que busca extraer información a partir de texto, es decir de datos no estructurados. En esta parte de la práctica, nos centraremos en las técnicas clásicas para representar un texto como un vector de frecuencias de aparición de las palabras que lo componen. Una vez representado el texto de forma estructurada, podemos aplicar técnicas estándar de minería de datos para analizar los textos, por ejemplo, clasificarlos o analizar la similitud entre dos textos.

Para alcanzar este objetivo, vamos a utilizar la herramienta RapidMiner con la que se pueden realizar procesos de minería de texto de forma rápida y eficiente.

OBJETIVO

El objetivo de la práctica es entrenar un clasificador que prediga la satisfacción de un viajero con su vuelo a partir de distintos datos del trayecto y de una reseña escrita por el viajero.

Para ello, utilizaremos una colección de reseñas sobre vuelos recogida por el autor Juhi Bhojani y publicada en Kaggle con una licencia OpenDatabase, que podéis descargar aquí

https://www.kaggle.com/datasets/juhibhojani/airline-reviews/data

El dataset contiene 23.000 reseñas

Las características que componen el dataset son las siguientes:

- Airline Name
- Overall Rating
- Review Title
- Review Date
- Verified (whether the review is verified or not)
- Review
- Aircraft
- Type of Traveller
- Seat Type
- Route
- Date Flown
- Seat Comfort
- Cabin Staff Service
- Food & Beverages

- Ground Service
- Inflight Entertainment
- Wifi & Connectivity
- Value for Money
- Recommended

El atributo a predecir es “Overall rating”. Este atributo tiene valores entre 1 y 9 y valores “n”. Descartaremos esos últimos. Para mejorar los resultados, si lo consideráis oportuno, podéis agrupar los valores en tres o cuatro niveles, siempre justificando la decisión.

Se recomienda que el preprocesamiento de los datos se haga con RapidMiner para que los resultados sean más fácilmente reproducibles.

Sesión I: Modelo básico sin usar texto

El objetivo de esta parte de la práctica es entrenar un modelo de clasificación o de regresión para predecir la etiqueta “Overall rating” a partir de los datos no textuales.

Podéis usar las herramientas “statistics” y “visualization” para hacer un análisis exploratorio de los datos, y decidir cuales incorporar y cuales dejar fuera (ventana resultados, pestañas laterales). Una alternativa recomendable para hacer el EDA es el operador “statistics”. Por otro lado, también debéis pensar en si merece la pena imputar los datos que falten y cómo hacerlo.

Sesión I: Modelo incorporando texto

1. Descarga de las extensiones de RapidMiner necesarias

La funcionalidad básica de RapidMiner puede ampliarse fácilmente con el uso de extensiones, que son módulos externos que realizan tareas concretas. Estas extensiones se descargan a través de menú Extensions > Marketplace (updates and extensions). La extensión que necesitarás inicialmente para esta práctica es Text Processing (para el pre-procesado de texto), que suele estar incluida en la instalación básica. Puedes comprobar si ya está instalada eligiendo Extensions > Manage Extensions.

2. Carga de texto en Rapid Miner

RapidMiner puede manejar texto de diferentes formas. Las principales son colecciones de documentos, que son documentos como objetos independientes dentro de un objeto (objeto “document collection”), y conjuntos de ejemplos en formato tabla, como los campos leídos de un fichero Excel (objeto “exampleSet”). Como veremos más adelante algunos operadores están disponibles en dos versiones para trabajar con cada tipo de datos. También existe un operador que convierte un “exampleSet” en “document collection” y viceversa.

En esta práctica vamos a trabajar con datos leídos de un csv, y cargados en un exampleSet.

La **configuración del operador que lee los datos de fichero es muy importante** para no tener problemas más adelante. Esta configuración se puede hacer usando el wizard "import configuration wizard" o directamente editando los metadatos como veremos a continuación.

La configuración de la carga del dataset permite establecer

1. El tipo de datos de cada campo del fichero: es muy importante comprobar que el texto de la reseña tiene asignado el tipo de datos texto, pues en caso contrario los operadores de manejo de texto que vamos a usar no funcionarán.
2. Qué campos del fichero se van a utilizar y cuales no son necesarios
3. Qué papel juega cada campo. En nuestro caso, el texto es un atributo (attribute), y la valoración (Overall rating) es una etiqueta (label), pues es el valor que queremos predecir.

Esta configuración del dataset se puede modificar con el botón **"dataset meta data information"** que lanza una ventana "pop up" para la configuración (*un aviso, este botón queda oculto cuando se visualiza RapidMiner en una ventana que no ocupa la pantalla completa, conviene tener cuidado con esto pues a veces no se ven todos los campos u opciones*).

Si preferís que esta configuración sea más visible en el diseño del proceso de Rapid Miner, podéis usar el operador "Nominal to Text" para convertir los atributos a tipo texto, el operador "Set Role" para definir el rol de los campos y el operador "Filter attributes" para conservar sólo los campos que queremos analizar y eliminar el resto.

Para comprobar si la carga de datos es correcta, conectamos la salida del operador al puerto res (resultados), y ejecutamos el proceso (icono Run). Si pasamos de la pestaña "Design" a la pestaña "Results", veremos los datos leídos en forma de tabla.

3. Análisis descriptivo de los datos

En este punto es recomendable hacer un análisis descriptivo de los datos, ver el número de reseñas, numero de reseñas por clase, si el conjunto está o no balanceado, las palabras más frecuentes etc. Una forma es utilizar la herramienta (visualizations) disponible en la pestaña de resultados. Algunos ejemplos de visualizaciones son un gráfico de barra en el que veamos cuantas entradas hay de cada tipo, o una nube de palabras (Word cloud) para visualizar las palabras más frecuentes.

4. Revisión y limpieza de los datos

Como estamos trabajando con datos reales, podemos a encontrar diferentes problemas e inconsistencia en los ejemplos.

Para empezar, es necesario establecer correctamente la codificación de los textos (por ejemplo, UTF-8) y revisar si no hay errores en este sentido.

Por último, puedes ser útil comprobar que todos los ejemplos están etiquetados, eliminando los que no están. Una opción es usar "Select examples"

También se puede usar "Sample" para quedarse con un subconjunto de los ejemplos.

5. Generación de la matriz de términos por documento

El siguiente paso es procesar el texto, siguiendo los pasos típicos del proceso de minería de texto, y crear la matriz de términos por documento en el formato que elijamos. En nuestro problema cada línea del fichero es un documento del corpus a efectos del procesamiento de la información.

Usaremos el operador "Process documents from Data" (Figura 1), que espera como entrada una tabla con los textos en formato "example Set" y da como salida, el texto procesado. Nótese hay otro operador equivalente, llamado "Process documents from file", que es el que deberíamos usar si nuestros textos están en documentos en un directorio en vez de en un Excel o csv.

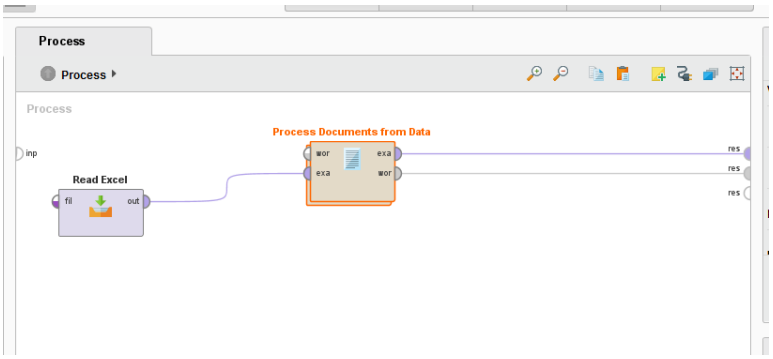

Figura 1. Procesar texto

Una vez añadido el operador "*Process documents from Data*", lo desplegamos (doble click) para insertar los operadores correspondientes (es decir, se insertan subprocesos dentro del proceso). Estos son los pasos más comunes de pre-procesado del texto que puedes añadir (Figura 2)

1. Identificación de los términos individuales (tokens) en el texto. Para ello, se utiliza el operador *Tokenize*.
2. Filtro de palabras que no son de interés (Stopwords). RapidMiner tiene incorporados varios filtros para palabras en inglés (Filter stopwords-English). Si más adelante queremos eliminar palabras concretas podemos crear con ellas un diccionario y eliminarlas añadiendo el operador Filter Stopwords (Dictionary).
3. Poner todo en minúsculas: Operador *Transform cases*

4. Eliminar palabras demasiado cortas : *Filter Tokens (By Length)* – el número de letras es configurable, normalmente no se conservan las palabras de dos o tres letras.
5. Reducir las palabras a su raíz. Para ello añadimos un operador que ejecute un algoritmo de *stemming,* el operador *Stem* con el algoritmo Snowball configurado para inglés.

En función de la tarea a realizar, algunos de estos pasos pueden no ser necesarios o ser contraproducentes. Por ejemplo, stem puede en algunos casos empeorar el resultado. Hay otras opciones de pre-procesado que pueden ser útiles, como eliminar palabras de un diccionario creado por nosotros.

Cuando terminemos esta parte probaremos a añadir o quitar algunos pasos del procesado de los textos y ver el efecto.

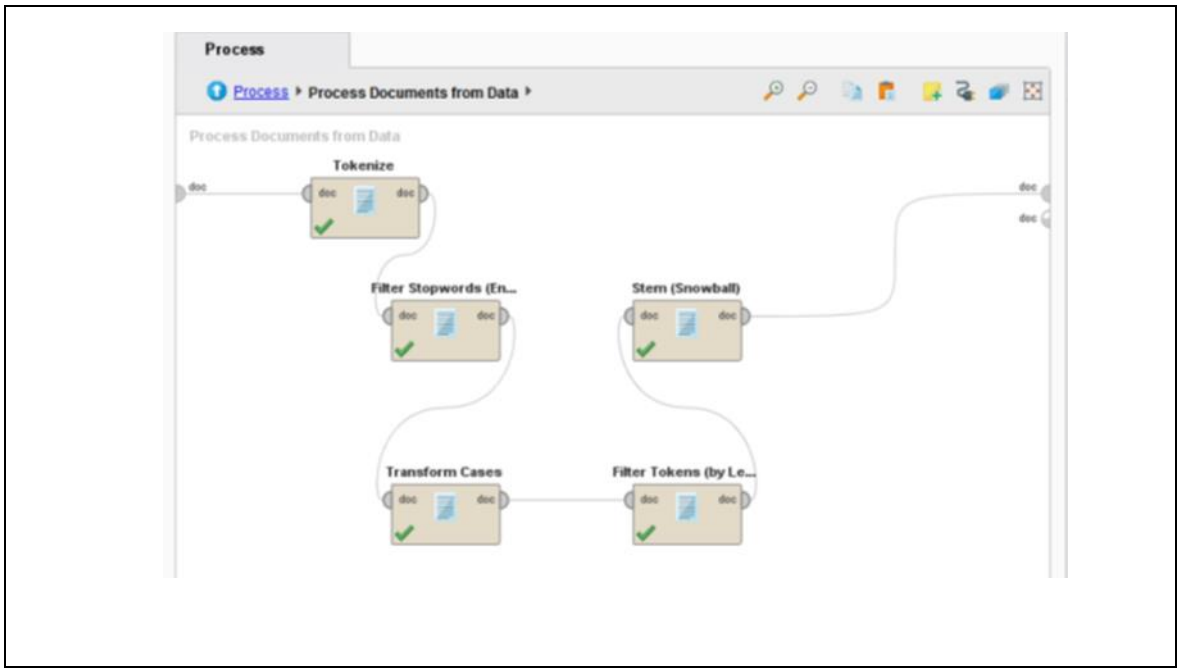

Figura 2. Ejemplo de sub-proceso englobado en el operador Process Documents from Data.

Volvamos a la pantalla principal de diseño (pulsando sobre el nombre del proceso o sobre la flecha que está junto al nombre) para configurar el operador “Process Documents from Data” y ver el resultado del proceso.

Como se puede ver en la Figura 1 el operador tiene dos puertos de salida exa y wor. En el primero se genera la matriz de términos por documento (tipo de datos Example Set) y en el segundo, una lista de las palabras del corpus con su frecuencia de aparición total y frecuencia de aparición en documentos (tipo de datos Word List). Conectaremos ambos a dos puertos de resultados.

Antes de ejecutar el proceso, estableceremos la configuración del operador “Process Documents from Data”. El principal parámetro a configurar es la forma de crear la matriz

de términos por documento, según cómo queramos que se represente la frecuencia de cada término: cuenta (term occurrence), frecuencia, frecuencia binaria (presencia vs ausencia) y tf-idf (frecuencia de término – frecuencia inversa del documento, term frecuency –inverse document frecuency). Estas opciones se establecen en el desplegable vector creation.

Comienza explorando la opción básica (cuenta: term occurrence), que es más fácil de interpretar. Prueba después con otras representaciones.

Además, el operador "Process Documents from Data" permite establecer un método de poda (prune) que limita los términos que se utilizan para generar la matriz de términos por documento, eliminando los muy frecuentes y poco frecuentes. En función de la tarea a realizar, este paso será útil o no. Por ejemplo, se puede establecer una poda ligada al porcentaje (percentual), haciendo que la cota inferior (prune_below_percent) sea el 3%, dejando la cota superior (prune_above_percent) al 100%, o reducir la cota superior para eliminar términos muy frecuentes.

Ejecutemos el proceso para inspeccionar los dos resultados: matriz de términos por documento (de tipo example set) y lista de palabras (de tipo word list). Veamos primero la word list. Si ordenamos las palabras por el número de veces que aparecen, podemos ver los términos más comunes, y cómo de comunes son para cada una de las clases por separado. ¿Crees que bastaría con esta cuenta para clasificar las reseñas? Si inspeccionamos el example set, veremos la matriz en la que para cada entrada y término tenemos el número de veces que aparece. Podemos comprobar que es una matriz dispersa (la mayoría de los términos son 0). Puedes explorar diferentes visualizaciones de la matriz términos por documento en la ventana de respuestas.

Ahora, repite la ejecución del proceso con otras representaciones de los documentos como term frequency y TF-IDF, y quizá con otros valores y métodos de pruning.

6. Construir y evaluar un clasificador

Una vez construida la matriz de términos por documento, los datos ya tienen estructura y por lo tanto se puede construir un modelo de clasificación con cualquiera de las técnicas que ya conocéis. Utiliza validación cruzada para los resultados.