Ejercicio Knime 2

Alberto Armijo Ruiz

Preprocesamiento utilizado en los diccionarios

En este apartado se explicarán los preprocesamientos específicos utilizados para los diccionarios SentiWordNet y SenticNet. También se explicará el preprocesamiento utilizado sobre el dataset de reviews de IMdb.

Base de datos de IMdb

Como preprocesamiento utilizado dentro de la base de datos de IMdb se ha convertido las reviews de películas a documentos; tras esto se ha seleccionado la columna que indica el sentimiento de dicho sentimiento y el documento en sí. Una vez tenemos estas dos columnas, se han preprocesados los documentos para eliminar símbolos de puntuación, números, palabras con pocos caracteres; para este ejercicio se han eliminado todas las palabras con menos de 3 letras, también se ha eliminado *Stop Word* y se ha convertido cada documento a minúsculas. Además de esto, se ha utilizado la mitad de las reviews que hay en el archivo por falta de RAM para realizar ciertas operaciones; para ello se ha utilizado un particionamiento estratificado por el sentimiento, para que ambas clases estén equilibradas.

SentiWordnet

En este apartado se explica el preprocesamiento utilizado para utilizar las palabras del diccionario *SentiWordnet*. Lo primero que se ha hecho ha sido leer el fichero con un nodo *File Reader*; tras esto se han utilizado la información de las columnas *Col2* y *Col3*, estas columnas contienen la polaridad positiva y negativa de cada palabra. Por ello, se ha utilizado un nodo *Java Snippet* para clasificar las palabras como positivas o negativas, para ello se calcula la diferencia entre las dos columnas, si la diferencia es positiva y mayor que 0.15, a una nueva columna llamada *Sent* se añade una entrada del tipo “positive”; si la diferencia es negativa y menor que -0.15, se clasifica como “negative” y si la diferencia se encontrara entre -0.15 y 0.15 entonces se clasifica como “neutral”.

Una vez se ha clasificado cada una de las palabras, se utiliza dos nodos *Row Filter* para seleccionar solamente las palabras clasificadas como “positive” y “negative” por separado y después se selecciona solamente la columna que contiene las palabras para después utilizarlas para generar los diccionarios.

SenticNet

El preprocesamiento realizado para el diccionario *Senticnet 5* ha sido parecido al realizado para *SentiWordnet*. Para este diccionario lo primero que se ha hecho ha sido utilizar un nodo *Java Snippet* para diferenciar entre palabras con intensidad alta de los que tienen poca intensidad; para ello, se utiliza el contenido de la columna *Intensity* y se compara si el

valor es menor que -0.2 o mayor que 0.2. Si es menor que -0.2, se considera que la palabra es negativa, si es mayor que 0.2, se considera que la palabra es positiva y si la intensidad está entre esas dos medidas, se considera que la palabra es neutral; esto se almacena dentro de una nueva columna.

Por último, al igual que se hacía para el diccionario *SentiWordnet* se separa la tabla en dos seleccionando las palabras de polaridad positiva y negativa con la columna que se ha creado antes y tras esto se selecciona únicamente la columna que contiene la palabra.

Resto del procesamiento

El resto de procesamiento utilizado para calcular el sentimiento que tiene cada documento ha sido muy parecido al que se proporciona en el ejemplo. Lo primero que se hace es añadir etiquetas a las palabras de los documentos que están dentro del documento usado. Tras esto, se realiza una puntuación por documento, para ello se crea una bolsa de palabra, se filtran todas las palabras que no hayan sido etiquetadas y se calcula la frecuencia por término. Tras esto, se realiza una agrupación por términos y se obtiene un valor positivo y negativo por dicho término; se procesan los valores pérdidos y se obtiene una medida global de cada término, calculada como la diferencia entre el peso positivo y negativo de cada término.

Lo siguiente es obtener una puntuación global por documento, para ello se agrupa la puntuación de cada término del documento y se obtiene la puntuación total del documento. Si dicha puntuación es mayor o igual a 0, se considera que el documento tiene un sentimiento positivo, sino tiene un sentimiento negativo. Por último, se comparan los resultados procesados con los sentimientos reales y se obtiene el rendimiento obtenido por el diccionario. También se utiliza un nodo *Color Manager* para después representar cada uno de los documentos según los valores de términos positivos y negativos.

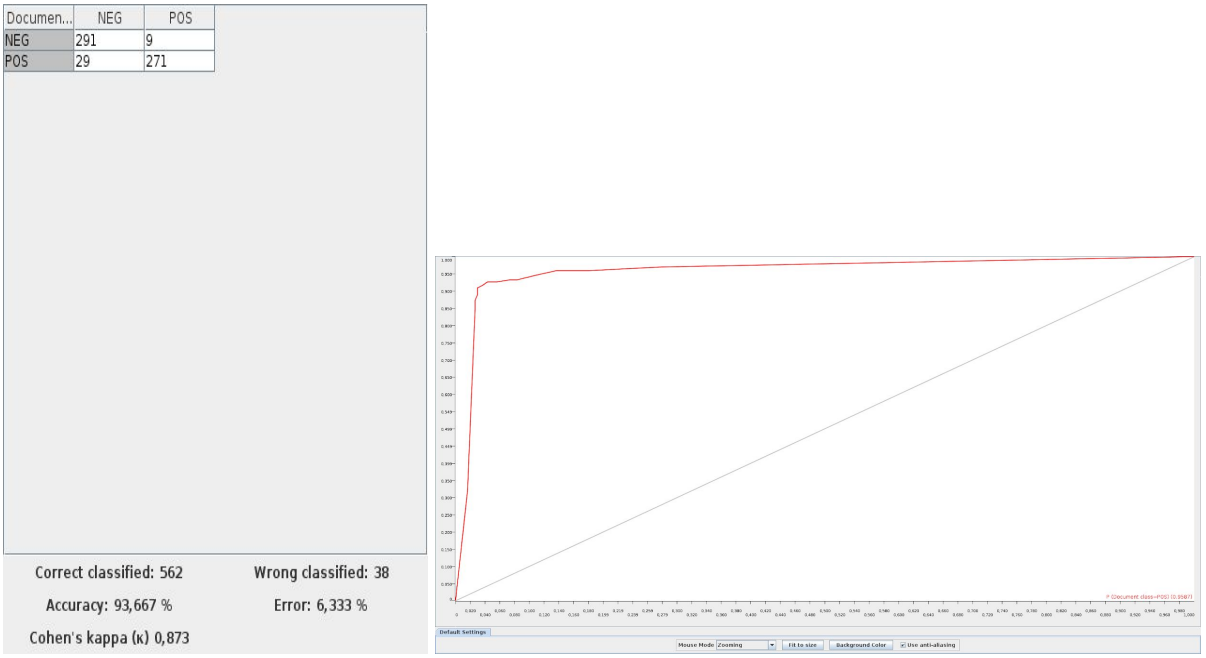
Preprocesamiento para la clasificación de sentimientos

El preprocesamiento seguido para la clasificación de sentimientos es muy parecido al utilizado en el apartado de análisis. Además de eliminar signos de puntuación, números, stop words y palabras de pocos caracteres; se realiza un steamming de los documentos y se crea una bolsa de palabras sobre estos. Una vez se han obtenido términos de los documentos de la bolsa de palabras, estos se pasan a strings y se calcula la frecuencia de los términos en los documentos y se seleccionan aquellos términos que aparezcan en al menos 20 documentos.

Tras esto se realiza se crea un vector de documentos donde se representan los términos seleccionados y su presencia o no en dicho documento. Por último, se realizan particiones para train y test y se obtienen los resultados.

Comparación de resultados

En este apartado se compararán los resultados obtenidos por los procesamientos de documentos con el rendimiento obtenido con algoritmos clásicos, en este caso los algoritmos utilizados son árboles de decisión y SVM. Los resultados obtenidos por los modelos son los siguientes.



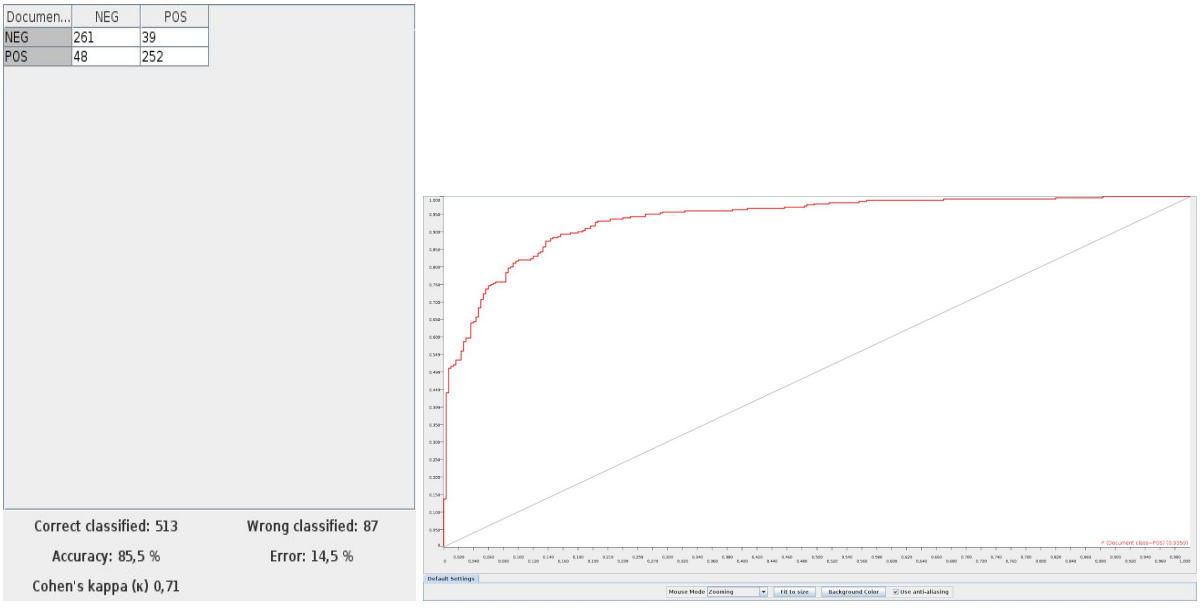
Documen...	NEG	POS
NEG	291	9
POS	29	271

Correct classified: 562 Wrong classified: 38

Accuracy: 93,667 % Error: 6,333 %

Cohen's kappa (κ) 0,873

Resultados obtenidos por árbol de decisión



Documen...	NEG	POS
NEG	261	39
POS	48	252

Correct classified: 513 Wrong classified: 87

Accuracy: 85,5 % Error: 14,5 %

Cohen's kappa (κ) 0,71

Resultados obtenidos por SVM

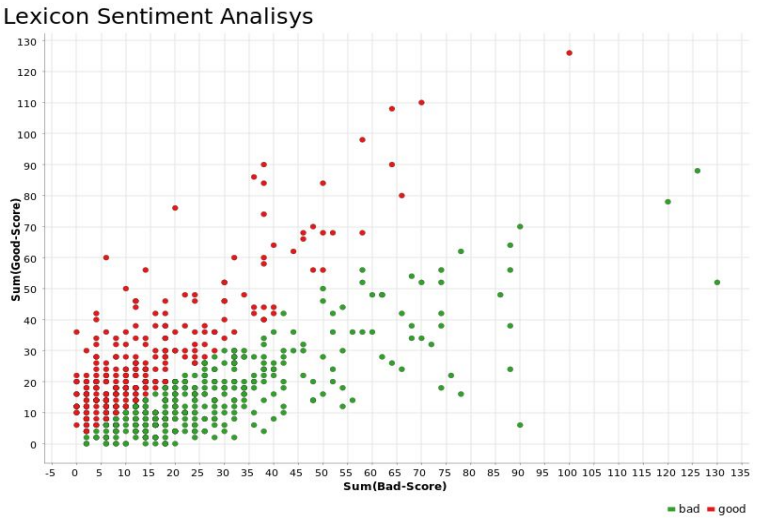
Los resultados obtenidos por los diccionarios son los siguientes.

Sentimen...	POS	NEG
POS	200	100
NEG	61	238

Correct classified: 438 Wrong classified: 161

Accuracy: 73,122 % Error: 26,878 %

Cohen's kappa (κ) 0,463



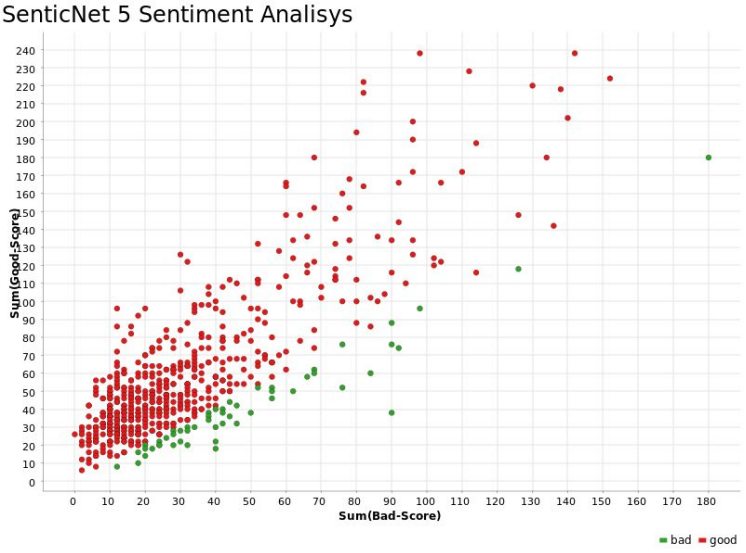
Resultados obtenidos por el lexicon

Sentimen...	POS	NEG
POS	288	12
NEG	259	41

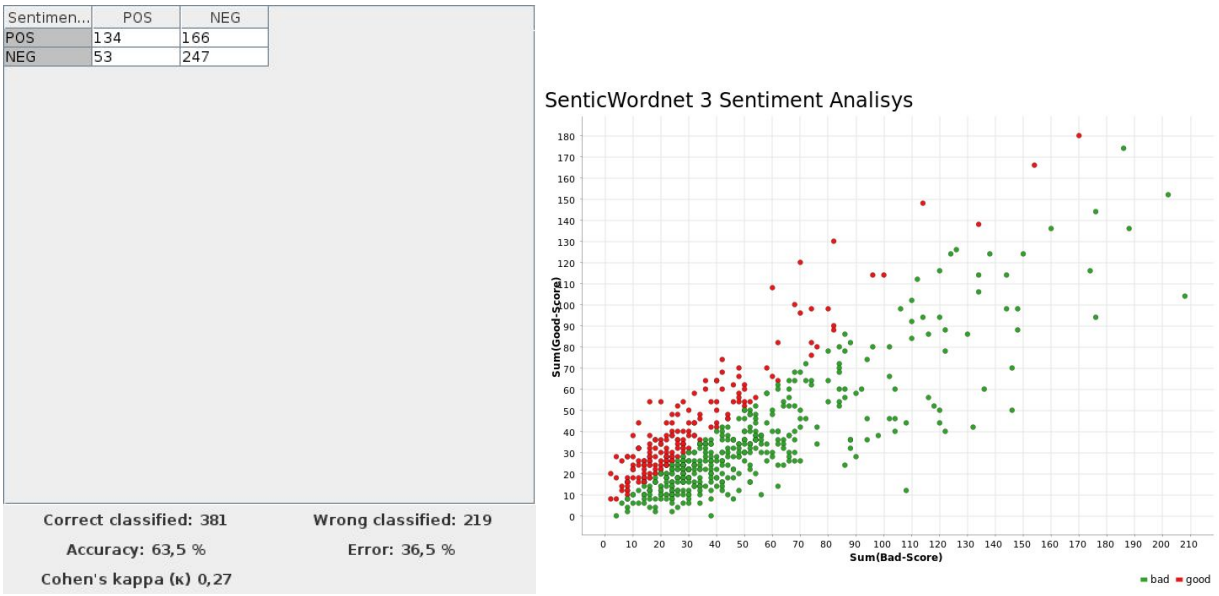
Correct classified: 329 Wrong classified: 271

Accuracy: 54,833 % Error: 45,167 %

Cohen's kappa (κ) 0,097



Resultados obtenidos por Senticnet

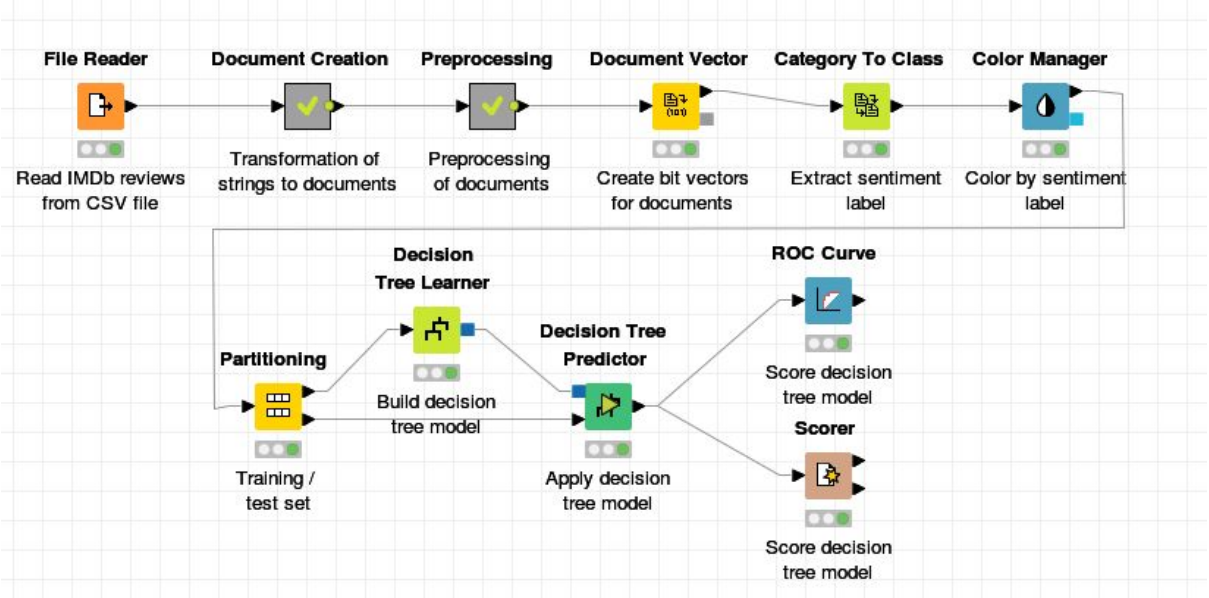


Resultados obtenidos por SentiWordNet

Como se puede ver en los resultados; los modelos clásicos obtienen bastante mejores resultados en comparación con el procesamiento utilizando los diccionarios. El modelo que mejor rendimiento obtiene son los árboles de decisión, aunque SVM puede ser capaz de obtener mejores resultados si se ajustan los parámetros de este modelo. Dentro de los resultados obtenidos por los diccionarios, el que mejores resultados obteniene es el lexicon *MPQA Subjetivity Lexicon*; que obtiene un porcentaje de acierto cercano al 75%, por lo cual se puede decir que relativamente hace un buen trabajo prediciendo el sentimiento de cada documento, sin embargo no es tan bueno como los modelos clásicos. Los otros dos diccionarios obtienen porcentajes bastante pobres que rondan el 60% de acierto o menos, aunque *SentiWordnet* obtiene mejores resultados que *SenticNet*, por lo cual no son adecuados para predecir los sentimientos de estos documentos, al menos con el procesamiento utilizado en la práctica; es posible que para *SenticNet* se pueda apreciar un aumento en la calidad de los resultados si se realizara un análisis sobre los *N-Gramas*.

WorkFlows Knime

WorkFlow Clasificación de sentimientos



WorkFlow Análisis de sentimientos