Proyecto: Text Mining. Support Vector Machines

sistemas de apoyo a la decisión

javier gil, guzmán lópez, javier suárez, Oliver Ye

2018

Tabla de contenidos

[**1.** **Introducción a la minería de textos** 2](#_Toc512683859)

[**2.** **Pre-procesamiento** 3](#_Toc512683860)

[Ejemplos de transformación 3](#_Toc512683861)

[Selección de atributos. Definición. 6](#_Toc512683862)

[**3.** **Clasificación** 6](#_Toc512683863)

[Baseline: Clasificador Naive Bayes 6](#_Toc512683864)

[Algoritmo asignado: Support Vector Machines 7](#_Toc512683865)

[**4.** **Marco experimental** 8](#_Toc512683866)

[Pruebas del software 10](#_Toc512683867)

[**5.** **Conclusiones** 10](#_Toc512683868)

[**6.** **Bibliografía** 10](#_Toc512683869)

[Páginas web consultadas 10](#_Toc512683870)

[Libros consultados 11](#_Toc512683871)

# **Introducción a la minería de textos**

La minería de datos la podemos definir como el análisis matemático para deducir patrones y tendencias que existen en los datos, patrones que no pueden detectarse mediante una exploración tradicional de los datos porque las relaciones son demasiado complejas o por el volumen de datos que se maneja.

Estos patrones y tendencias se pueden recopilar y definir como un modelo de minería de datos. Se puede decir que *Data Mining* se refiere al conjunto de métodos estadísticos que proporcionan información (correlaciones o patrones) cuando se dispone de muchos datos.1

En el campo de la inteligencia artificial, la minería de textos se aplica en el Procesado del Lenguaje Natural. Pero esta contiene multitud de subdivisiones: Comprensión del Lenguaje Natural, Generación de Lenguaje Natural, construcción de Conocimiento Base, Sistemas de Gestión del Diálogo, Procesamiento del Habla…etc.

Estas disciplinas permiten, por ejemplo, que un sistema no humano aprenda un lenguaje de forma automatizada, utilizarlo de forma relativamente coherente o adquiera ciertos conocimientos derivados de estos textos.2

**Aplicaciones2:**

-*Social Media Analytics* (SMA) consiste en la recogida de datos de las redes sociales (Twitter, Facebook, blogs, RSS, páginas web) distribuidos en diversas fuentes, tras realizar un análisis automatizado de la información, estos son mostrados en un entorno gráfico para ayudar a los empresarios a tomar decisiones empresariales.

-IBM *Watson* es una plataforma tecnológica que utiliza el Procesado de Lenguaje Natural para obtener patrones y resúmenes de grandes cantidades de información no estructuradas. *Watson* es capaz de extraer información de todo tipo de documentos de texto enriquecido de un repositorio para entonces construir patrones sobre las relaciones de los distintos conceptos de los documentos.

**Retos3:**

Uno de los retos que debe afrontar la minería de textos frente a la minería de datos tiene que ver con el análisis semántico de textos, ya que este análisis resulta computacionalmente muy costoso, pues los sistemas únicamente son capaces de operar en el orden de unas pocas palabras por segundo.

Otro retro es el análisis de textos multilenguaje. Los sistemas actuales están, en su mayoría, centrados en el inglés. Sería interesante ver cómo podrían tratar estos sistemas otros idiomas o incluso textos compuestos por varios idiomas.

# **Pre-procesamiento**

Características más relevantes de los distintos tipos de datos:

-Raw. Estos datos no están en ningún formato concreto, es la representación de los datos tal cual son obtenidos en su fuente.

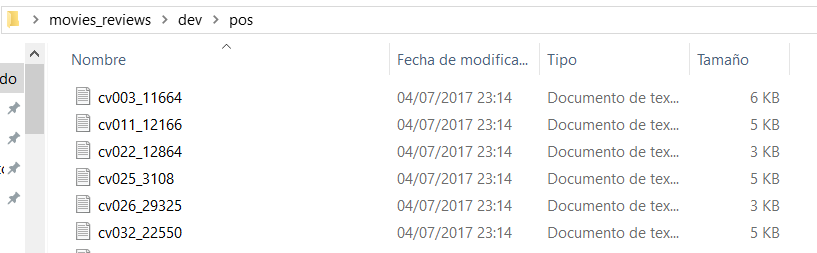
-BoW (*Bag of Words*). Este formato es una representación del texto que describe la ocurrencia de las palabras en un documento o directorio. Utiliza un vocabulario de palabras conocidas y una medición de la presencia de las palabras conocidas.

-TF-IDF (*Term Frequency-Inverse Document Frequency*). Es una estadística numérica que refleja la importancia de unas ciertas palabras en el contexto de un documento o directorio.

## Ejemplos de transformación

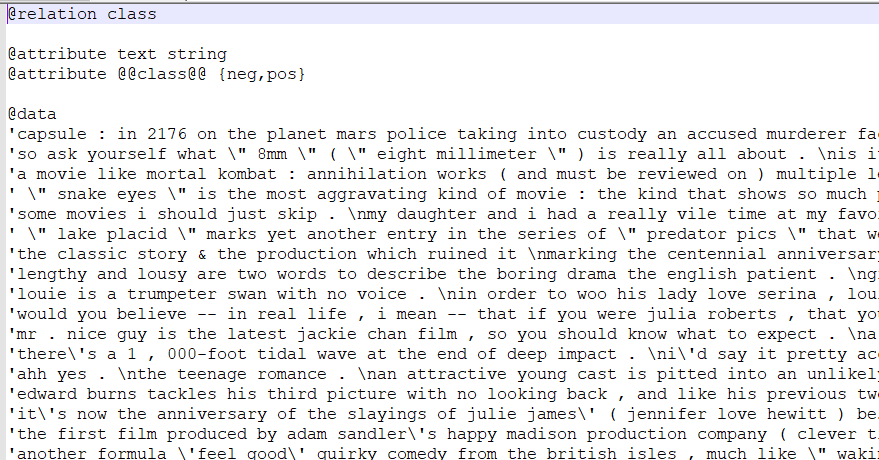
Para probar el funcionamiento de las aplicaciones, utilizaremos como datos de ejemplo “Movies reviews”, que nos han sido proporcionados para la realización de este proyecto. “Movies reviews” está compuesto por una colección de publicaciones sobre opiniones de películas otorgadas por unos 500 usuarios, los cuales han sido clasificados en negativos y positivos en referencia a críticas desfavorables y favorables, respectivamente.

El conjunto de críticas clasificadas como negativas están situadas dentro de la carpeta “neg”, mientras que las positivas están en “pos”. Además, estos datos están separados en 3 subconjuntos: train, dev y test (clase desconocida). Para este ejemplo utilizaremos el subconjunto dev. Cada crítica está guardada en un fichero .txt dentro de las carpetas mencionadas:

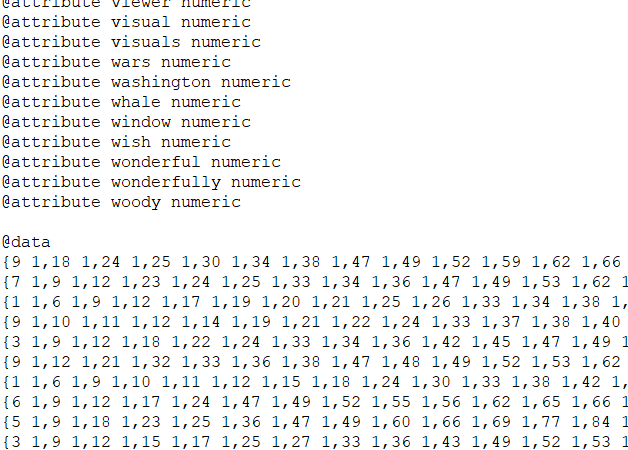


-Convertir a formato Raw:

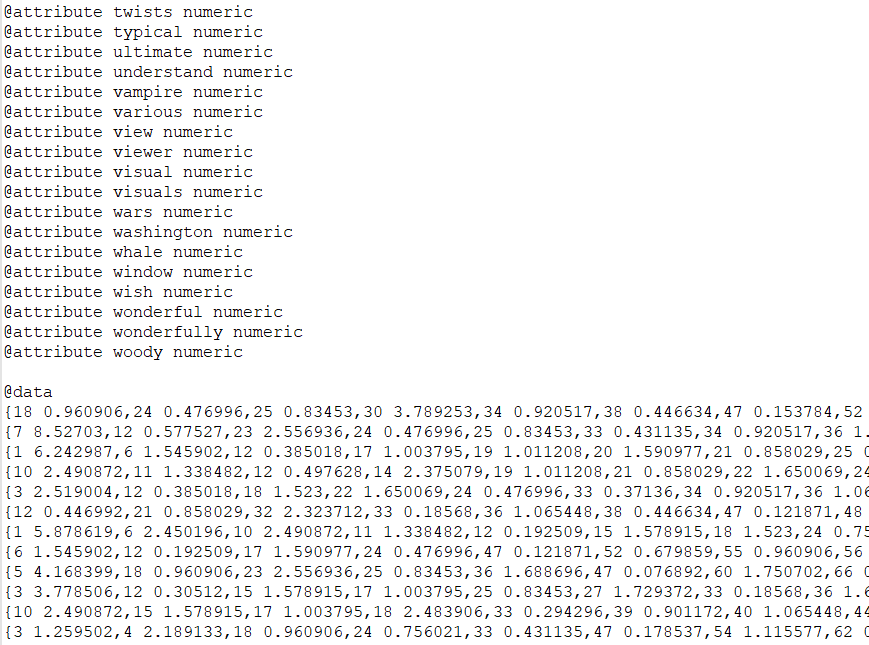
Una vez convertido el directorio en un fichero .arff, su contenido queda de la siguiente manera:



-Transformar del formato Raw a BoW:



-Transformar del formato Raw a -IDF:



En cuanto a la fase práctica de esta tarea, hemos utilizado la plataforma GitHub en conjunción con Eclipse para el desarrollo del software. Hemos consultado toda la información relevante sobre los métodos y clases de Weka y Java en la Wiki oficial de Weka ( <https://weka.wikispaces.com/Use+WEKA+in+your+Java+code> ).

La repartición de las tareas está distribuida de la siguiente manera:

|  |  |
| --- | --- |
| Tarea (Fase 1 del proyecto) | Autor |
| Parte 1: Representación raw. Implementación y pruebas. | Guzmán López |
| Parte 2: Transformación (BoW, TF-IDF). Implementación y pruebas. | Javier Gil |
| Parte 3: Hacer compatible. Implementación y pruebas. | Oliver Ye |
| Informe | Javier Suárez |

|  |  |
| --- | --- |
| Tarea (Fase 2 del proyecto) | Autor |
| Parte 1: FSS Software | Oliver Ye |
| Parte 2: Obtener modelo baseline y su calidad estimada. | Guzmán López |
| Parte 3: Optimización de parámetros. | Javier Gil |
| Parte 4: Obtener modelo predictor SVM. | Javier Suárez |
| Informe | Todos |

|  |  |
| --- | --- |
| Tarea (Fase 3 del proyecto) | Autor |
| Software predictor. | Guzmán López, Javier Gil, Javier Suárez |
| Informe | Oliver Ye |

## Selección de atributos. Definición.

La selección de atributos consiste en, teniendo un conjunto de datos, extraer aquellos atributos que sean realmente relevantes y que mejoren la calidad del modelo, dado que entre esos datos pueden existir datos incompletos o redundantes, que no aportan nada en cuanto a la calidad del modelo.

# **Clasificación**

## Baseline: Clasificador Naive Bayes

Como algoritmo *baseline,* es decir, el umbral inferior de lo que se espera obtener en la tarea, tenemos el modelo Naïve Bayes (algoritmo de Bayes ingenuo).

El algoritmo de clasificación Naïve Bayes es un clasificador probabilístico. Se basa en modelos de probabilidades que incorporan fuertes suposiciones de independencia. Estas suposiciones de independencia no tienen ningún efecto sobre la realidad, por lo tanto, se consideran ingenuas (naïve en inglés)4.

Este algoritmo funciona de la siguiente manera:

Dado un ejemplo representado por valores, el clasificador Naïve Bayes se basa en encontrar la hipótesis más probable que describa a ese ejemplo. Si la descripción de ese ejemplo viene dada por los valores la hipótesis más probable será aquella que cumpla:

es decir, la probabilidad de que, conocidos los valores que describen a ese ejemplo, éstos pertenezcan a la clase (donde es el valor de la función de clasificación perteneciente al conjunto finito V). Por el teorema de Bayes:

Podemos estimar contando las veces que aparece el ejemplo en el conjunto de entrenamiento y dividiéndolo por el número total de ejemplos que forman este conjunto. Para estimar el término , es decir, las veces en que, para cada categoría aparecen los valore del ejemplo x, debo recorrer todo el conjunto de entrenamiento. Este cálculo resulta impracticable para un número suficientemente grande de ejemplos por lo que se hace necesario simplificar la expresión. Para ello se recurre a la hipótesis de independencia condicional con el objeto de poder factorizar la probabilidad. Esta hipótesis dice lo siguiente:

*Los valores que describen un atributo de un ejemplo cualquiera son independientes entre sí conocido el valor de la categoría a la que pertenecen.*

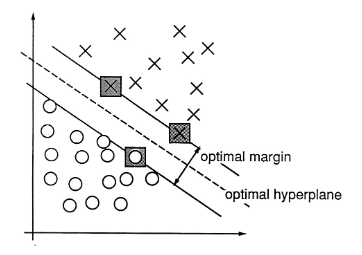
Así, la probabilidad de observar la conjunción de atributos dada una categoría a la que pertenecen es justamente el producto de las probabilidades de cada valor por separado: . [Mitchell, 1997, Chap. 6]

## Algoritmo asignado: Support Vector Machines

Los Support Vector Machines (SVM) fueron desarrollados por Cortes y Vapnik (1995) [Cortes & Vapnik, 1995] para la resolución de problemas de clasificación y regresión. Dado un conjunto de muestras de entrenamiento es posible etiquetar las clases y entrenar una SVM para construir el modelo que prediga la clase de una nueva muestra.

Una SVM es un modelo que representa las instancias de la muestra en un espacio multidimensional basado en los siguientes conceptos:

**Separación de clases:** básicamente, se intenta buscar el hiperplano separador de dos clases óptimo maximizando el margen entre las instancias más cercanas de ambas clases, estas instancias que residen en la frontera se llaman *support vectors* (vectores de soporte en inglés), y en el centro de este margen se encuentra el hiperplano separador óptimo:



*Figura 1. Ejemplo del margen óptimo entre 2 clases. [Cortes & Vapnik, 1995, Fig. 2]*

**Clases solapadas:** a las instancias situadas en el lado “incorrecto” del margen discriminatorio se les otorga una menor ponderación para reducir su influencia (“soft margin”).

**No linearidad:** cuando no es posible encontrar un separador lineal, las instancias son proyectadas en un espacio de mayor dimensión donde las instancias son efectivamente separables linealmente mediante un hiperplano. Esta proyección se realiza mediante métodos de *Kernel.*

**Solución de problemas:** toda esta tarea puede ser formulada como un problema de optimización cuadrática que puede ser resuelto utilizando técnicas conocidas.

Por lo tanto, para que este clasificador sea lo más eficaz posible, las muestras deberían:

* Estar muy claramente clasificadas, ya que este es un algoritmo binario, principalmente.
* No contener mucho ruido ya que generan solapamiento de clases.
* Ser de tamaño no muy grande, pues el tiempo de entrenamiento puede ser elevado.

# **Marco experimental**

Antes de comenzar con el desarrollo, a continuación, mostramos el funcionamiento de cada paquete de software:

1. Adquisición y pre-procesamiento de datos:
2. Inferencia del clasificador:
3. Predicciones

//TO DO

## Pruebas del software

Para probar nuestro software, hemos realizado varias pruebas con los conjuntos de datos proporcionados para la realización de este proyecto. Aquí se muestra un ejemplo utilizando el conjunto de datos “Tweet Sentiment”:



# **Conclusiones**

# **Bibliografía**

## Páginas web consultadas

[1] Revista Digital INESEM

¿Qué es y cuáles son las aplicaciones del Text Mining?

<https://revistadigital.inesem.es/informatica-y-tics/text-mining/> (consultado el 27/03/18)

[2] UniRioja

Text Analytics: the convergence of Big Data and Artificial Intelligence (Documento PDF)

<https://dialnet.unirioja.es/descarga/articulo/5573981.pdf> (consultado el 27/03/18)

[3] IEEE

Text Mining: Challenges and Future Directions (Documento PDF)

<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7124872> (consultado el 27/03/18)

[4] IBM

Clasificación Naive Bayes

<https://www.ibm.com/support/knowledgecenter/es/SSEPGG_9.7.0/com.ibm.im.overview.doc/c_naive_bayes_classification.html> (consultado el 14/04/18)

## Libros consultados

[Mitchell, 1997] Mitchell, Tom (1997). *Machine learning.* McGraw-Hill.

[Cortes & Vapnik, 1995] Cortes, C. & Vapnik, V. (1995). *Support-vector network. Machine Learning.* Kluwer Academic Publishers.