Análisis del sentimiento en Twitter

El análisis de sentimiento o “sentiment analysis” en redes sociales es una tendencia muy actual y con diversas aplicaciones.

Debido a que el clasificador Naive Bayes ha sido utilizado con éxito para detectar mensajes Spam, tiene sentido utilizarlo también en la detección del análisis de sentimiento en los tweets.

En está practica se solicita al alumno/a crear un filtro detector del sentimiento de los tweets utilizando el software R. En primer lugar se describen los pasos necesarios para construir el conjunto de datos. A continuación se deja libertad al alumno/a para completar el trabajo y se valora la creatividad y exhaustividad del mismo.

Conjunto de datos

Vamos a utilizar un conjunto de datos de tweets con 180 ejemplos, de los cuales 90 ejemplos corresponden a tweets que indican alegría (happy) y 90 a tweets que indican tristeza (sad). Este conjunto se ha obtenido de Twitter con los hashtags de #Happy y #Sad. Se ha dividido en conjuntos de train y test y se puede descargar del blackboard.

Cada línea del conjunto de datos contiene el texto del tweet correspondiente al sentimiento del fichero, por ejemplo:

I am LOVIN my Life right about now! I'm loving the people God is placing in my life. #Happy&Focused! Striving to be the BEST WOMAN I can be!

#iLikePeopleWho makes me laugh when I am sad.

Preparación de los datos

Trabajar con texto puede resultar complicado porque hay que transformar las palabras y sentencias a una representación que pueda ser comprendida por un ordendador. En nuestro caso vamos a utilizar un paquete de R llamado **RTextTools** que proporciona las funciones comunes para trabajar con texto.

En primer lugar comenzamos leyendo el conjunto de datos del disco:

```
setwd("/home/fede/Dropbox/utad/Grado/Grado-2015-2016/DM/practicaNB/")
happy.train <- readLines("happyTrain.txt")
sad.train   <- readLines("sadTrain.txt")
happy.test  <- readLines("happyTest.txt")
sad.test    <- readLines("sadTest.txt")
```

A continuación vamos a concatenar los datos:

```
tweets.train <- c(happy.train, sad.train)
tweets.test  <- c(happy.test, sad.test)
tweets.all    <- c(tweets.train, tweets.test)
```

Ahora creamos las etiquetas para los tweets anteriores:

```
class.train <- c(rep("happy", length(happy.train)), rep("sad", length(sad.train)))
class.test  <- c(rep("happy", length(happy.test)), rep("sad", length(sad.test)))
class.all   <- as.factor(c(class.train, class.test))
```

El siguiente paso consiste en crear la matriz con la aparición de las palabras en cada uno de los tweets para poder entrenar el clasificador Naive Bayes. Para ello vamos a utilizar la función **create_matrix()** del paquete **RTextTools, e**l cual es necesario instalar y cargar en el proyecto.

```
mat = create_matrix(tweets.all, language = "english", removeStopwords = FALSE,
                    removePunctuation = FALSE,
                    removeNumbers = TRUE, stemWords = FALSE, tm::weightTfIdf)

mat = as.matrix(mat)
dim(mat)
View(mat)
```

Entrenando un modelo sobre los datos

Ahora que ya tenemos los datos en un formato que puede ser uilizado pora un modelo de aprendizaje automático se solicita al alumno/a que cree este modelo utilizando el subconjunto de datos para entrenamiento y realizar la predicción sobre el conjunto de test.

Es necesario utilizar la función naiveBayes() del paquete e1071 para entrenar el modelo y la función predict() para realizar las predicciones sobre el conjunto de validación.

Asímismo se solicita probar con distintos tamaños de conjuntos de entrenamiento y validación y comentar los resultados obtenidos de forma exhaustiva. Por ejemplo, ¿cuales son los tweets que indican *sad* pero son clasificados como *happy*.

También es necesario realizar pruebas cambiando los diferentes parámetros de la función create_matrix(). Se valorará también la ampliación del conjunto de entrenamiento y validación.

La práctica se evaluará de la siguiente forma:

1) **(2 puntos)** Ejecución de los pasos de preparación de datos hasta ejecutar el modelo.

2) **(4 puntos)** Creación del modelo Naive Bayes.

3) **(4 puntos)** Creatividad aplicada a la solución. Experimentación con los parámetros de la función create_matrix() y como afectan a los resultados. Explicación de los resultados experimentales obtenidos. Se valoraran los comentarios debidadmente justificados y la claridad de los mismos así como lo riguroso y exhaustiva que sea la validación del modelo.

Normativa de realización, entrega y evaluación de la práctica:

- La práctica se realizará y entregará por grupos compuestos de 2 personas como máximo.
- La práctica se realizará en R/RStudio, utilizando el software instalado en el aula o bien una versión idéntica.
- La entrega deberá hacerse mediante Blackboard antes del 29 de noviembre de 2015 a las 23:59 horas (hora peninsular en España).
- La entrega se compondrá de un único fichero ZIP, que contendrá lo siguiente:
    a. Un fichero de texto llamado “prac_1_Nombres_Grupo.txt” que contendrá el código R necesario para realizar la práctica.
- **En ningún caso se debe entregar el modelo entrenado o variables guardadas a disco.**
- Al hacer entrega del fichero comprimido, éste se habrá renombrado con el nombre y el primer apellido de los alumnos integrantes del grupo, escritos en letras mayúsculas y separados mediante un guión bajo ‘_’. Ejemplo: JOSE_FONT_FEDERICO_CASTANEDO.zip
- La corrección de la práctica se hará ejecutando los ficheros proporcionados de manera secuencial y evaluando el resultado obtenido.
- Cualquier sospecha de copia entre dos o más prácticas derivará en la calificación de 0 para todos los alumnos involucrados.