Clasificación de texto: Deteción de humor mediante el uso de perceptrón simple y multicapa.

Julia Patrycja Wojciechowska

UPV/EHU (December 11, 2022)

Objetivos

El objetivo de este ejercicio es obtener el mejor clasificador posible empleando diferentes parámetros o variaciones de un algoritmo que clasifica un grupo de datos.

Dataset

El dataset está dividido en dos clases: humor o no humor. El atributo de cada instancia tiene formato de textos cortos y estos, están escritos en inglés.

Text	Humor
Divorce gift card: mexican ...	False
More like president pajama ...	True

Table 1:Formato de los datos

El dataset consta de 200 mil instancias y está equilibrado.

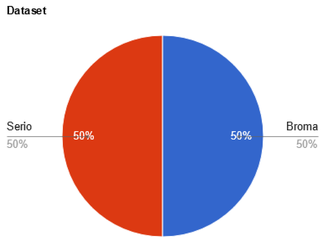

Figure 1:Proporción del número de instancias cada clase en el dataset

Preproceso de datos

1. Cambio de todas las letras mayúsculas a minúsculas
2. Lematización
3. Eliminación de stop-words
4. Tf-idf : Cambio de atributo 'Text' a atributos numéricos.

Baseline

Los datos representados en 2D tienen este aspecto

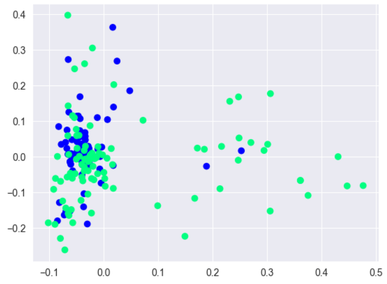

Figure 2:Representación de los datos en 3D

Mediante un **perceptrón** intentaremos trazar una recta para separar las dos clases.

Train	Test	Atributos
260	40	1 atributo aleatorio

Table 2:Parámetros del baseline

El **F-Score** obtenido es de tan solo **0.31**. Los siguientes experimentos servirán para mejorarlo.

Experimento nº 1: Perceptrón simple mejorado

Una recta no es suficiente; hay que emplear más atributos y crear hiperplanos. Para evitar overfitting se limita el número de iteraciones

Train	6400
Test	1600
Nº de atributos	50, 100, 200, 250, 300, 350
Nº de iteraciones	100, 250, 500, 1000, 2000

Table 3:Parámetros del experimento nº 1

¡Aplicando combinaciones de atributos e iteraciones, se llegó a conseguir en algunos casos un **F-Score** de **0.81**!

Tendencia: Cuanto mayor número de atributos e iteraciones, mejor resultado.

Experimento nº 2: Perceptrón multicapa

Si con un perceptrón simple conseguimos un 0.81, ¿qué pasará si empleamos un **perceptrón multicapas**?

Train	8000
Test	2000
Nº de atributos	200, 250, 300, 350
Nº de iteraciones	500, 1000, 1250, 1500, 2000
Nº capas ocultas	2, 4, 6, 8, 10
Alpha	0'32, 1, 3'16

Table 4:Parámetros del experimento nº 2

Alpha es un parámetro de regularización empleado para mejorar la generalización.

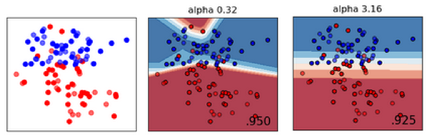

Figure 3:Regularización de alpha

F-Score ha aumentado hasta un **0.84**.

Tendencia: El resultado 0.84 mayoritariamente aparece con valor de alpha 1 y 8 capas ocultas del perceptrón.

Experimento nº 3: Perceptrón multicapa con ReLU

En el último intento de mejora del **F-Score**, se empleó la función de activaciom **ReLU**.

Los parámetros empleados fueron los mismos que los que se pueden observar en la tabla 4, con la diferencia de que no se empleó el valor 3'16 del parámetro alpha.

F-Score máximo conseguido: ¡**0.87**!

Modelo final

Para elegir el modelo ganador, se empleó las curvas ROC de cada clasificador que obtuvo un **F-Score** de **0.87**

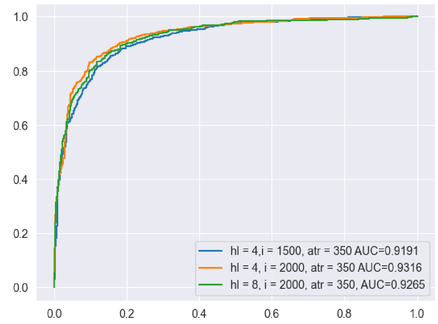

Figure 4:Curvas ROC de los mejores modelos

El modelo que más AUC posee tiene los siguientes parámetros:

Nº de atributos	350
Nº de iteraciones	2000
Nº capas ocultas	4
Alpha	1

Table 5:Parámetros del mejor modelo

Conclusiones

- Encontrar las mejores combinaciones de parametros requiere experimentación.
- Utilizar más capas ocultas de las que se necesita, puede llevar a sobreajsute.
- Con un algoritmo tan simple como un **Perceptrón**, los resultados obtenidos ya eran buenos.
- Como se puede observar en la Figura 2, hay bastantes instancias dificiles de separar, lo que podría ser la causa de lo poco que se consigió mejorar el **F-Score** a partir del primer experimento.