Práctica 5: Filtrado de Spam usando Bayes Ingenuo

1 - Objetivos

1. Entender cómo funciona un sistema de filtrado de spam en correos electrónicos.
2. Aplicar un clasificador de Bayes ingenuo a un problema real.
3. Diseñar e implementar un sistema de detección de spam en Python.
4. Evaluar el sistema implementado en bases de datos públicas.

2 - Estudio previo

Tanto en la práctica 5 como en el trabajo práctico TP6-2 se utilizará Python dentro del entorno Google Colab. El estudio previo consiste en familiarizarse con este entorno y la sintaxis básica de Python.

2.1 Python

El entrenamiento del clasificador se realizará mediante Python y se usará el paquete para aprendizaje automático scikit-learn[1]. Dicho paquete tiene dependencias con las librerías de cálculo científico NumPy y SciPy. Como entorno de trabajo vamos a utilizar Google Colab[2], editor online proporcionado por Google que permite la ejecución de scripts de Python sin necesidad de instalar ningún programa en nuestro ordenador. Recordamos que UNIZAR tiene un convenio con Google por el que se dispone de cuenta en Google Drive con espacio ilimitado si utilizamos nuestro NIP como usuario[3].

Si no estás familiarizado con Python ni Google Colab te recomendamos que antes de asistir a la práctica abras y ejecutes el siguiente tutorial.

Prueba a descargar una copia del cuaderno dentro de tu Google Drive para poder editarla y ejecutarla sin problemas.

---

[1] http://scikit-learn.org/stable/
[2] https://colab.research.google.com/
[3] https://sicuz.unizar.es/correo-y-colaboracion/espacios-web-colaborativos-inicio/espacios-web-colaborativos-documenta-y-g

3 - Desarrollo de la práctica

3.1 Bases de datos públicas

Para la resolución del trabajo se usará la base de datos pública Enron-Spam[4], diseñada para el entrenamiento y evaluación de sistemas de filtrado de spam. En concreto se usarán los correos electrónicos preprocesados (denominados Enron1, Enron2,… Enron6).

Se proporciona un notebook de Colab[5] como punto de partida para el desarrollo de la práctica. La primera celda del notebook carga los paquetes de Python que se utilizarán en la práctica. La segunda celda de código incluye las instrucciones para descargar y descomprimir automáticamente estos ficheros en las 6 carpetas. El entrenamiento y validación del clasificador se realizará con los correos de Enron1 a Enron5, y el conjunto Enron6 se reservará como datos de test.

3.2 Entrenamiento y evaluación de un clasificador de Bayes Ingenuo

El notebook proporcionado incluye el código básico para construir y entrenar un clasificador de SPAM basado en Bayes Ingenuo con distribución de Bernoulli y suavizado de Laplace, k=1. También se incluye código de ejemplo para el cálculo de las principales métricas utilizadas para analizar la calidad del clasificador utilizando los datos de test.

Analiza el código proporcionado, consulta la documentación de Python para entender los parámetros de las principales funciones e interpreta los resultados que obtiene el clasificador.

3.3 Ejercicios a realizar

3.3.1 Elección de la mejor configuración de clasificador

En esta parte de la práctica el objetivo es elegir la mejor configuración de clasificador posible. Para ello en primer lugar divide los datos de entrenamiento en dos subconjuntos, datos para entrenar y datos para validar, en una proporción 80-20 ó 90-10. El primer conjunto se utilizará para entrenar las diferentes variantes de clasificador y el segundo dato se utilizará para evaluarlo y comparar los resultados.

En la práctica se pide que entrenes las siguientes configuraciones:

1. Prueba el funcionamiento del clasificador de Bayes Ingenuo de distribución Bernoulli y el de distribución Multinomial.
2. Compara los resultados de ambos clasificadores para diferentes valores del hiper parámetro del suavizado de Laplace. Ten en cuenta que es posible poner tanto valores mayores como valores menores que 1.
3. Analiza cómo influye en el clasificador la utilización de bi-gramas como características en la bolsa de palabras.

[4] http://www2.aueb.gr/users/ion/data/enron-spam/
[5] https://colab.research.google.com/drive/1LQtJ86nzNnCmAy8ZNbw0AJeRuf6S3iSF?usp=sharing

Para determinar el mejor clasificador se utilizará el F1-score, obteniendo su valor a través de la predicción en los datos de validación.

4. OPCIONAL. Realiza el proceso de elección del mejor clasificador utilizando el algoritmo K-fold de evaluación cruzada.

3.3.2 Evaluación del mejor clasificador en los datos de test

Una vez que hayas elegido el clasificador y los hiper parámetros que mejor funcionan, la última parte de la práctica se va a centrar en el análisis más detallado de los resultados utilizando los datos de test y las diferentes métricas vistas en clase: F1-Score, matriz de confusión y curva precisión-recall.

1. Interpreta los resultados obtenidos para tu clasificador, ¿consideras que es un buen clasificador de correo SPAM?
2. Selecciona un umbral adecuado para el clasificador de spam en base a la curva precisión-recall, justificando la respuesta.
3. Selecciona de entre los correos electrónicos de test algunos ejemplos de spam y ham clasificados correcta e incorrectamente, y discute los resultados.

3.4 Documentación a entregar

Puedes descargar el notebook en tu Google Drive y editarlo, añadiendo las secciones de código y de texto que necesites para añadir comentarios y explicaciones. El propio cuaderno será tu informe de la práctica. Puedes añadir tantas secciones de código y de texto como consideres necesario para resolver todos los ejercicios propuestos y analizar los resultados obtenidos. Una vez hayas terminado, descarga el notebook en formato ipynb y súbelo a Moodle en la tarea habilitada para la P5 con el nombre NIP_P5.ipynb