PARCIAL I

COMPUTACIÓN BLANDA

DANIEL DIAZ GIRALDO

DANIEL FELIPE MARIN

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA, RISARALDA

Metodología:

1) limpieza de datos
    a) Adecuación para trabajar con pandas (python) - limpieza sintáctica.
2) Generalización del dataset
    a) Adecuación del proceso del paso 1 para dar una salida coherente (dataset matricial con características) ó label data.
3) Algoritmo Regresión logística:
    a) Aplicación de algoritmo para el dataset procesado por los pasos anteriores.
4) Toma de datos y evaluacion conceptual.
    a) Resultados.

Pasos realizados:

<u>Para algoritmo de clasificación emails.</u>

- Sustracción correcta del dataset de correos aplicando técnicas de bash.
  Se realizan en conjunto técnicas propias de sistemas unix (sed - tr) eliminando saltos de línea (\n) y retornos de carro (\r), luego se concatena cada línea en un archivo nuevo, importante eliminar los de todos los emails (;) dado que será nuestro delimitador y su importancia no tiene peso sobre el algoritmo.

- Generalización de dataset:
    - Usando pandas extraemos los archivos generados como cvs, dado esto, podemos facilitar la carga de [labels(0 ó 1);Ids(email.xxx);Email (.....)]
    - Para los labels e ids se proceden a cambiar el valor de “Spam” con 1 Y de “Ham” con 0. Por último agregamos un delimitador (;) para controlar el flujo del parsel en python.
    - Concatenamos archivos generados de los pasos a y b en uno nuevo, simulando un formato csv.
    - Limpieza por BeautifulSoup (api especializada en parseo de expresiones del tipo html)
    - Limpieza de caracteres fuera del rango de la expresión regular "[^a-zA-Z]" y espacios
    - Limpieza de stopwords
    - Generación de Características a partir de repetición en palabras.
    - Generación de archivo en archivo.

- Algoritmo de regresión logística
    - Optimizamos la carga a octave generando un archivo nuevo guardado con instrucciones nativas.
    - Separamos Aleatoriamente con permutaciones el porcentaje de datos para entrenar y para demostrar aprendizaje.

Para algoritmo de clasificación de reviews

- Generalización de dataset.
    - Usando pandas extraemos los archivos generados como cvs, dado esto, podemos facilitar la carga de [labels(0 ó 1);Ids(email.xxx);Email (.....)]
    - Para los labels e ids se proceden a cambiar el valor de “Spam” con 1 Y de “Ham” con 0. Por último agregamos un delimitador (;) para controlar el flujo del parsel en python.
    - Concatenamos archivos generados de los pasos a y b en uno nuevo, simulando un formato csv.
    - Limpieza por BeautifulSoup (api especializada en parseo de expresiones del tipo html)
    - Limpieza de caracteres fuera del rango de la expresión regular "[^a-zA-Z]" y espacios
    - Limpieza de stopwords
    - Generación de Características a partir de repetición en palabras.
    - Guardado en archivo.

- Algoritmo de regresión logística
    - Optimizamos la carga a octave generando un archivo nuevo guardado con instrucciones nativas.
    - Separamos Aleatoriamente con permutaciones el porcentaje de datos para entrenar y para demostrar aprendizaje.
    - Ver código.

Para ambos algoritmos se generó una función de evaluación que determina si el resultado puede ser clasificado como una muestra exitosa o no.

Características del ordenador:

```
hpc@arwen:/$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 58
Stepping:              9
CPU MHz:               1600.000
BogoMIPS:              7020.33
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              8192K
NUMA node0 CPU(s):     0-7
```

```
hpc@arwen:/$ lsblk -d -o name,rota
NAME ROTA
sda     0
hpc@arwen:/$
```

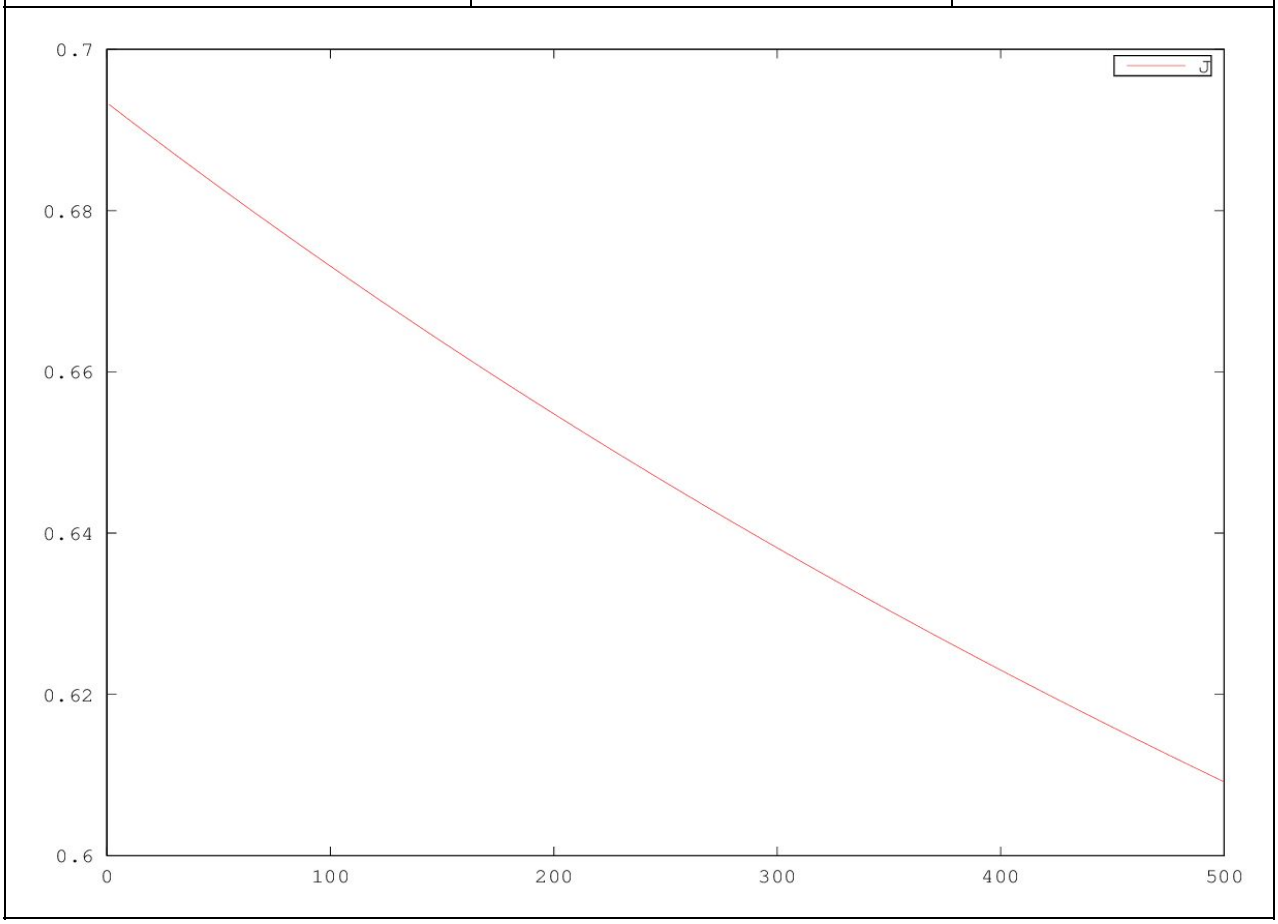
ssd

Resultados

Para el algoritmo clasificador de reviews:

Tamaño dataset	Muestras	Cracterísticas	Datos para entrenamiento (80%)	Datos para pruebas de factibilidad (20%)
25000 X 5000	25000	5000	20000	5000

Tiempo (segundos)	Alpha	N# Iteraciones
263.801	0.1	500



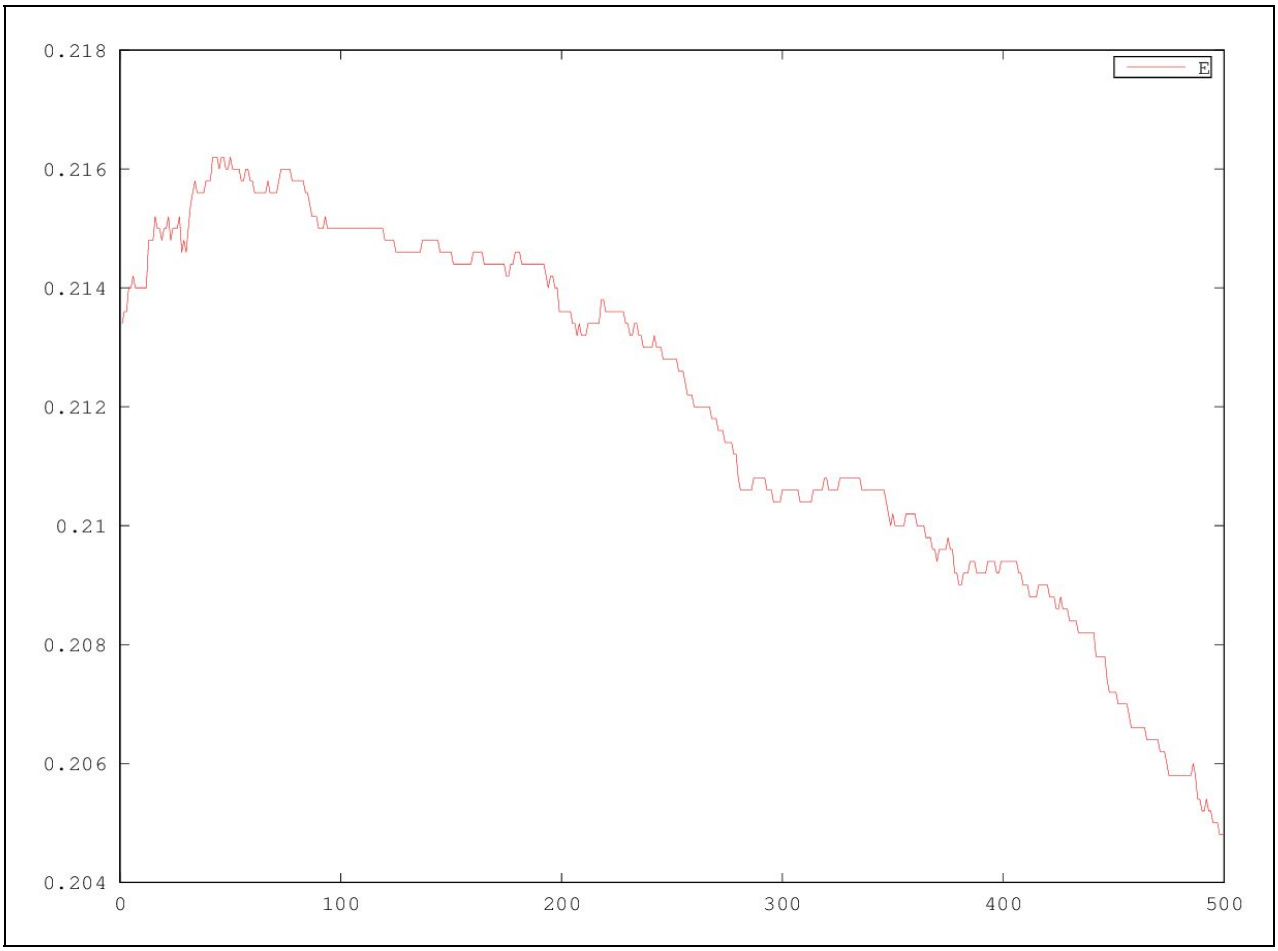
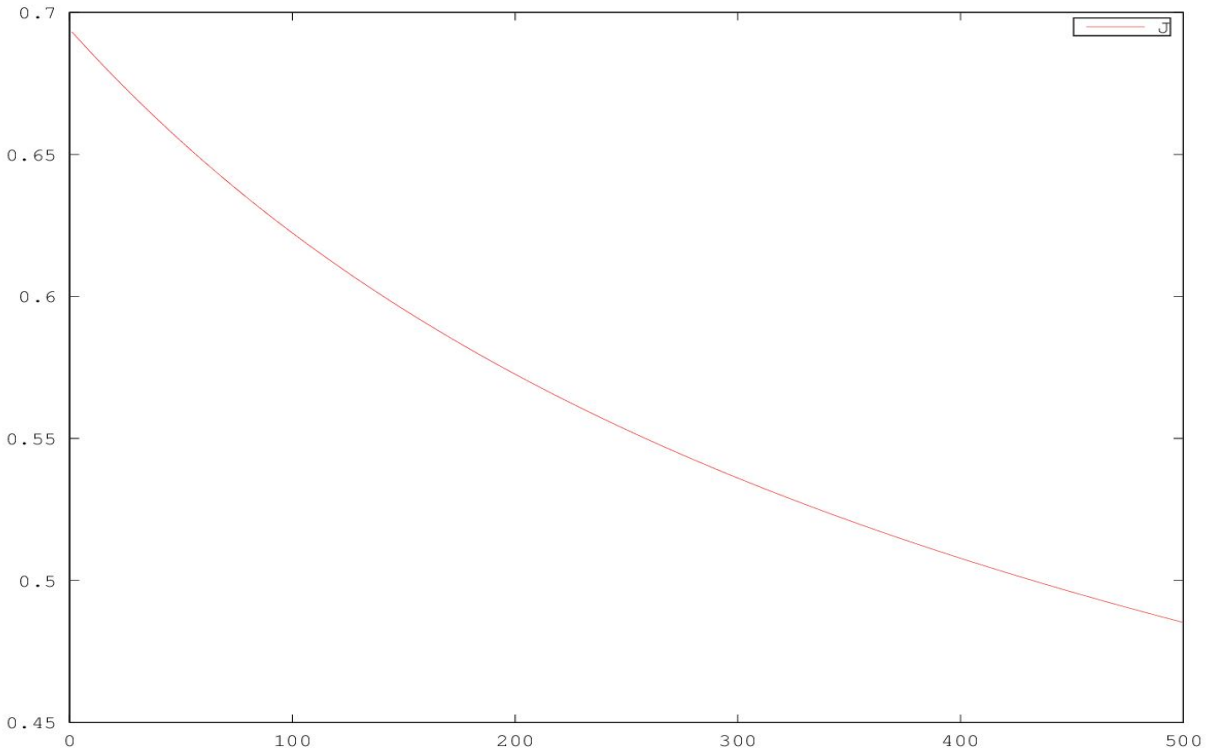


Imagen 1 - Resultados algoritmo clasificador reviews IMDB

Tiempo (segundos)	Alpha	N# Iteraciones
253.614	0.4	500



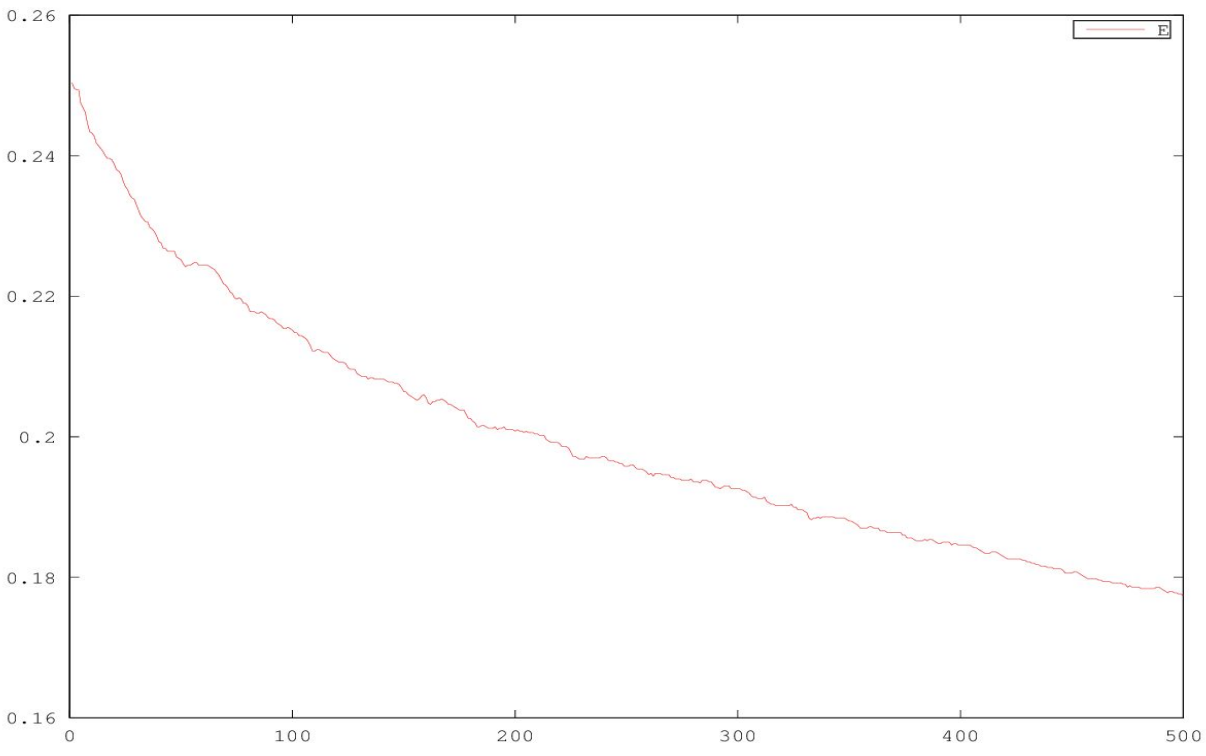
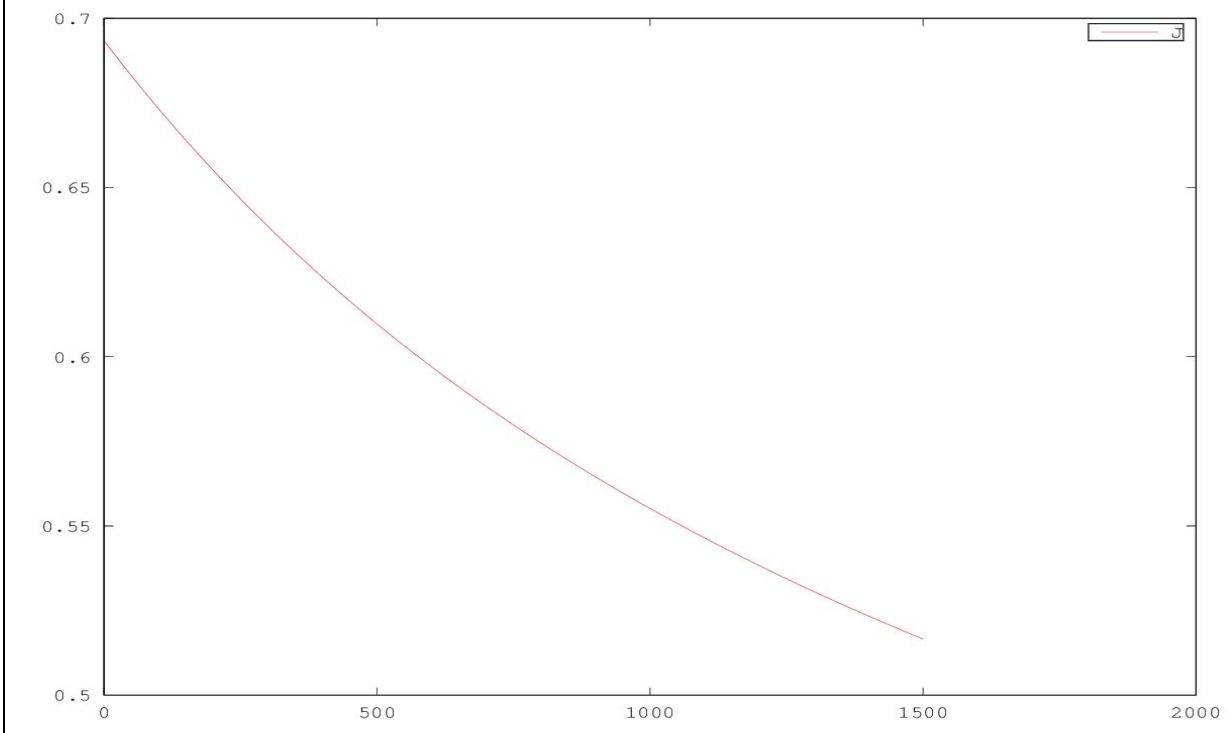


Imagen 2 - Resultados algoritmo clasificador reviews IMDB

Tiempo (segundos)	Alpha	N# Iteraciones
757.833	0.1	1500



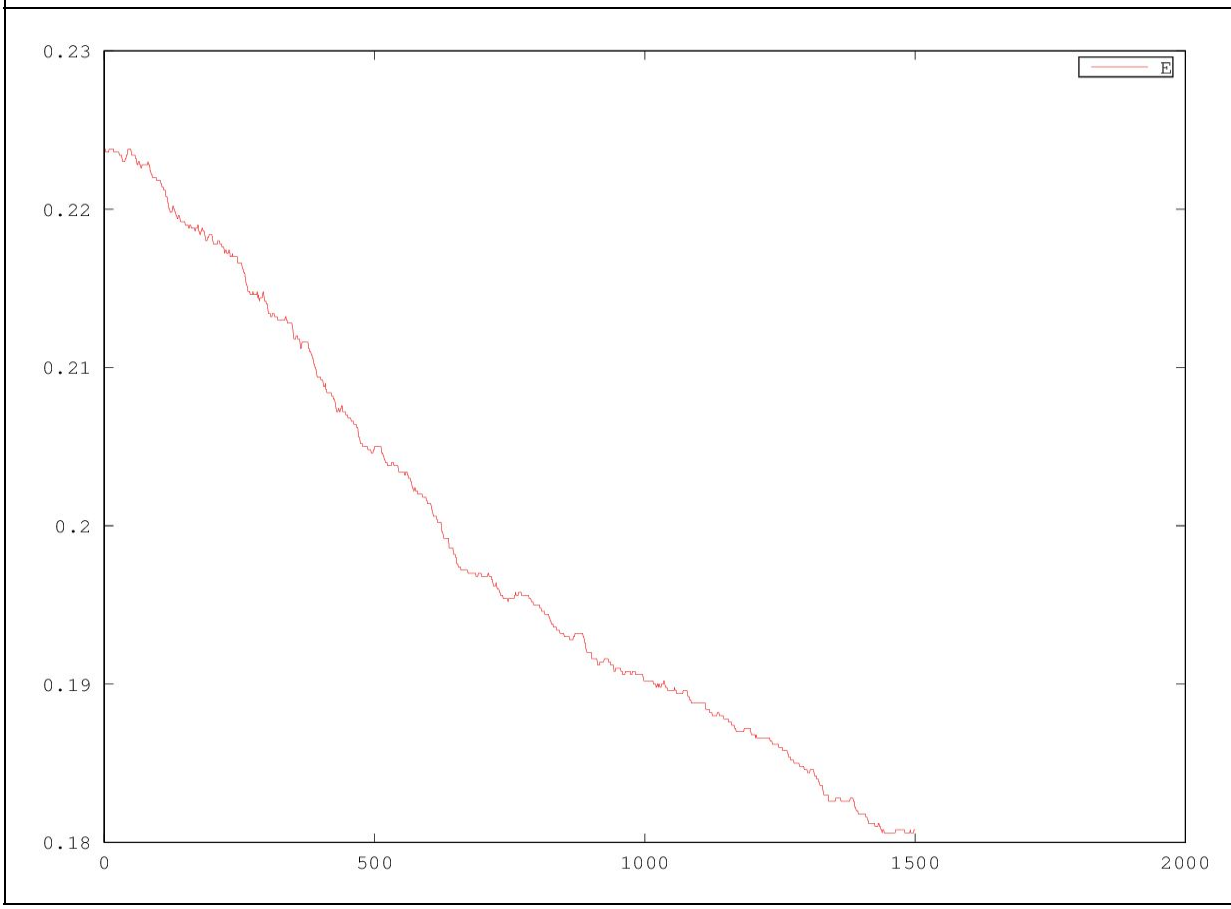


Imagen 3 - Resultados algoritmo clasificador reviews IMDB

Tiempo (segundos)	Alpha	N# Iteraciones
770.701	0.4	1500

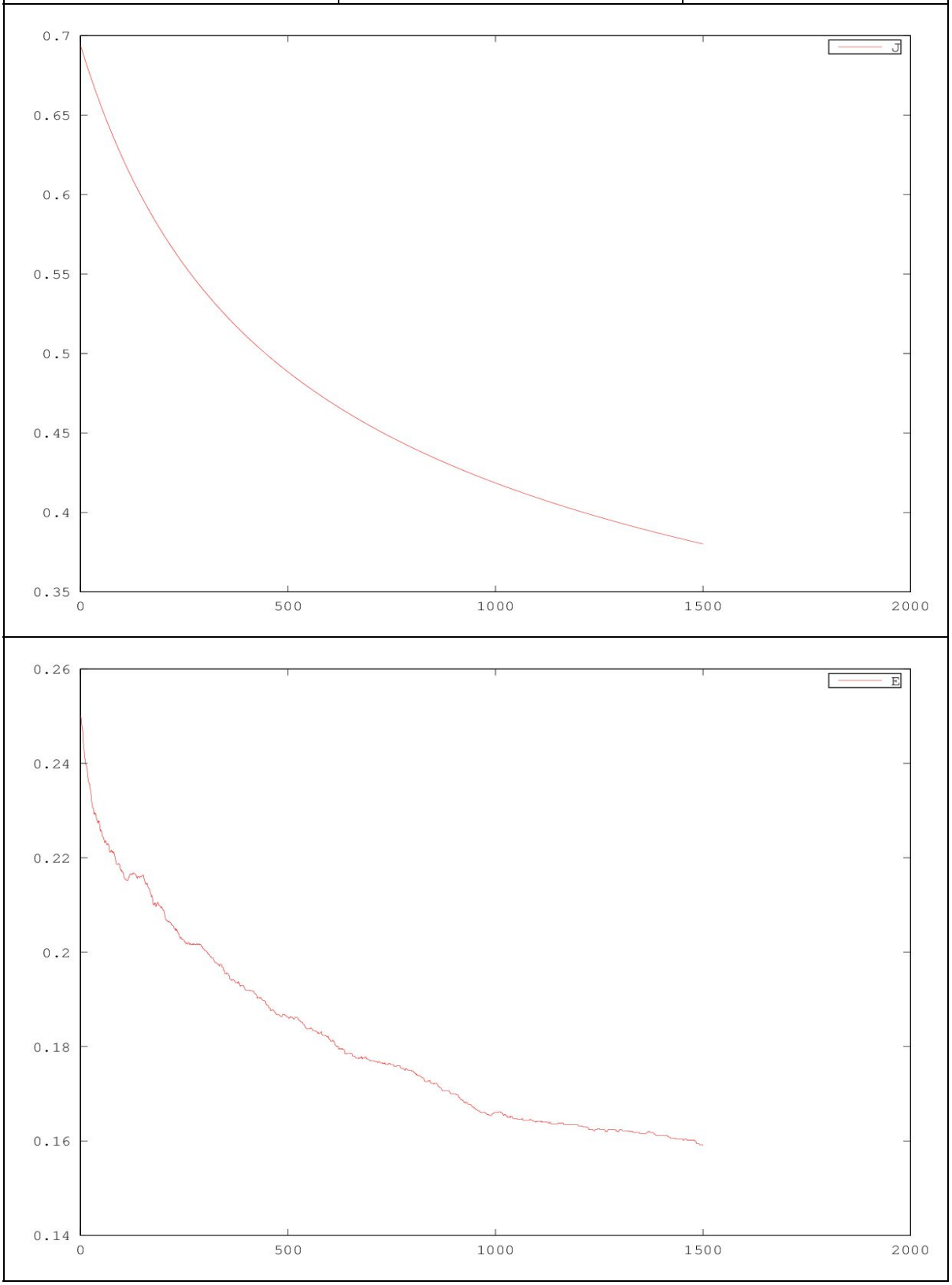


Imagen 4- Resultados algoritmo clasificador reviews IMDB

Tiempo (segundos)	Alpha	N# Iteraciones
1021.41	0.4	2000

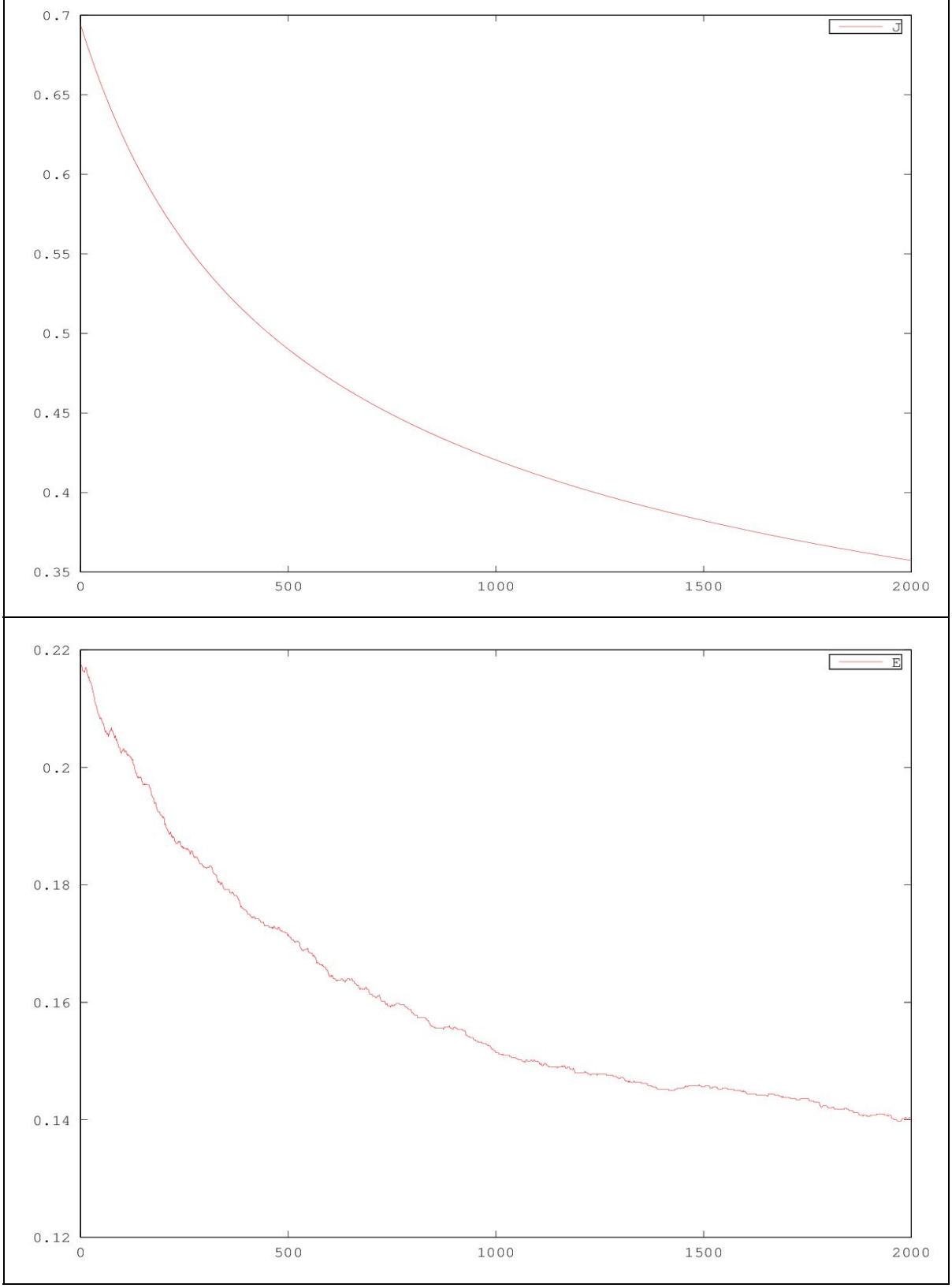
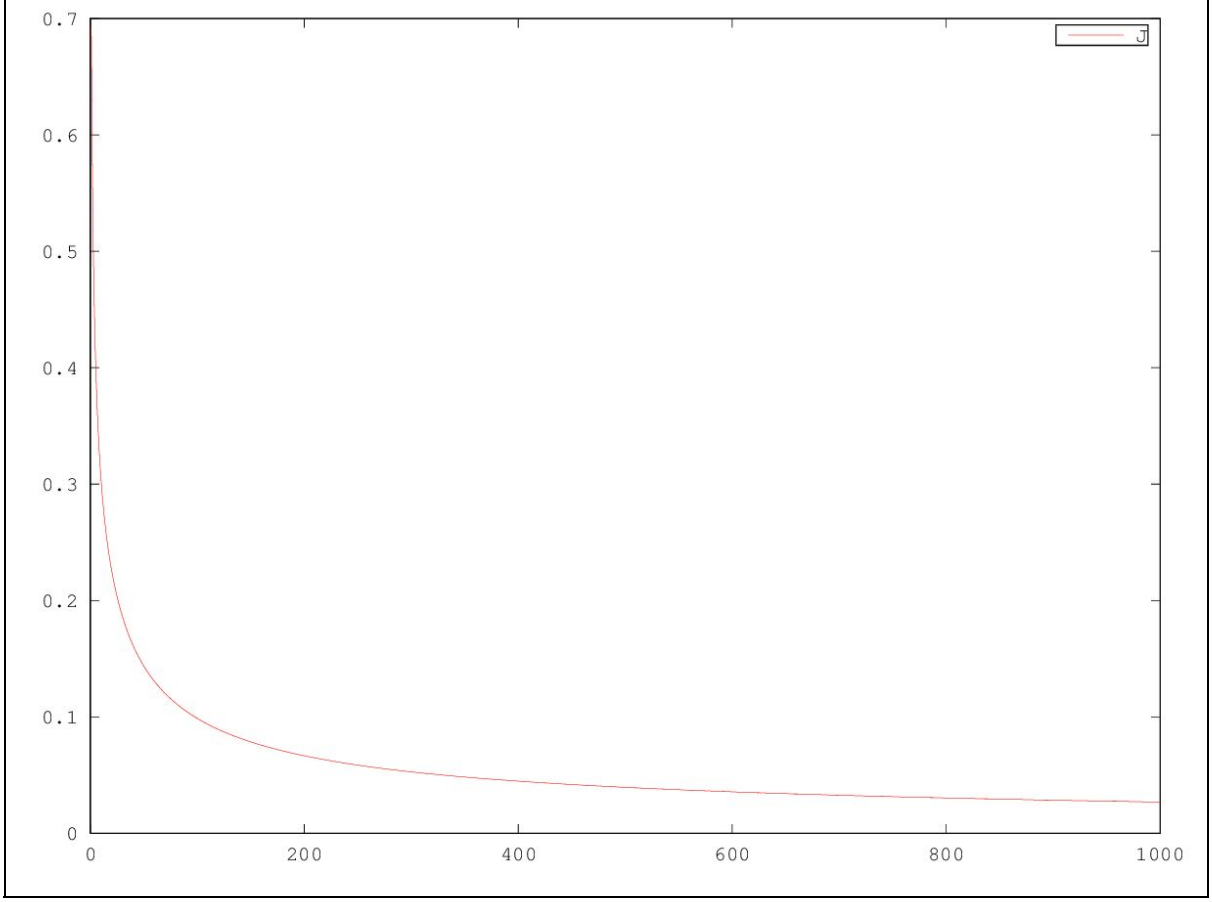


Imagen 5- Resultados algoritmo clasificador reviews IMDB

Para el algoritmo clasificador de emails

Tamaño dataset	Muestras	Características	Datos para entrenamiento (80%)	Datos para pruebas de factibilidad (20%)
75419 X 5000	75419	5000	60335	15084

Tiempo (segundos)	Alpha	N# Iteraciones
1453.44	0.01	1000



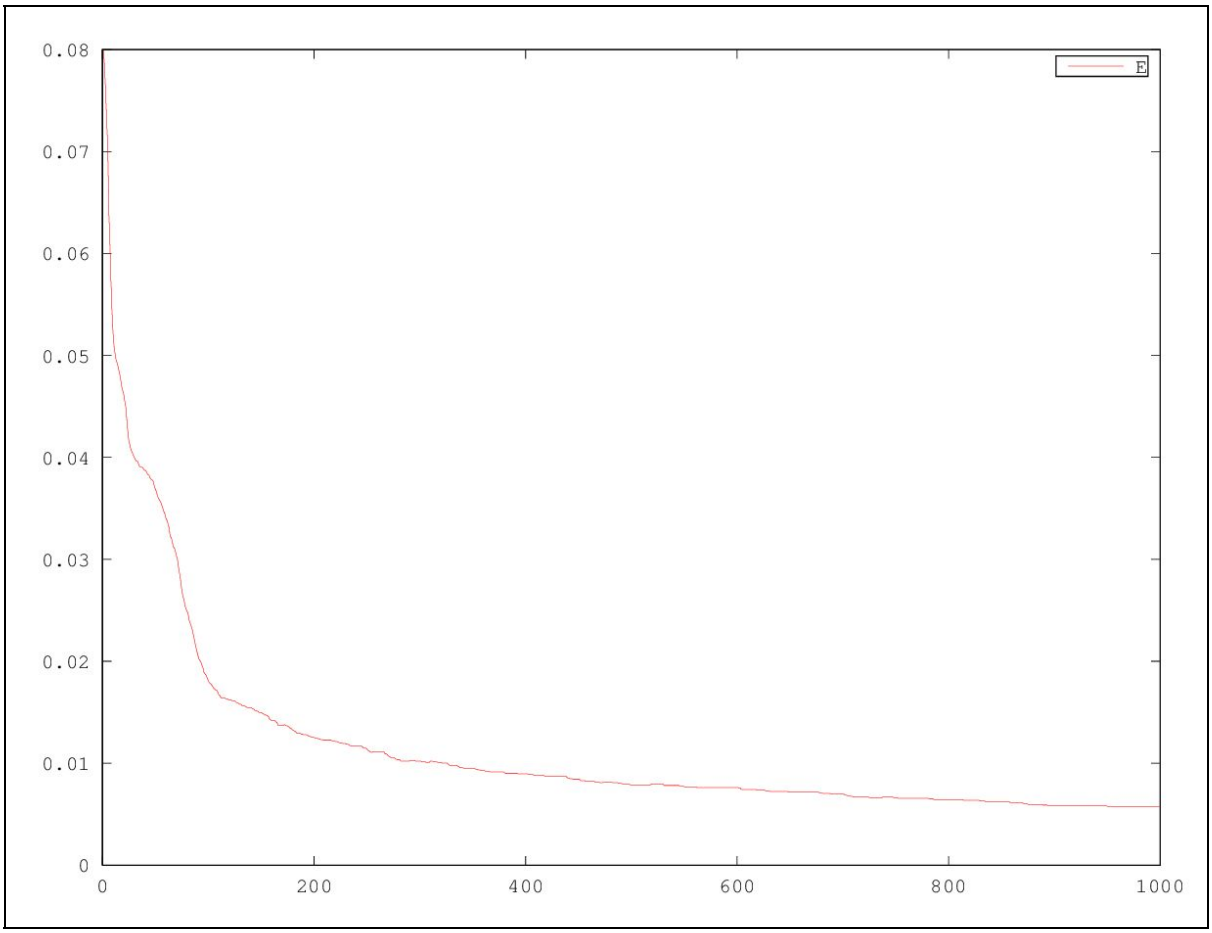
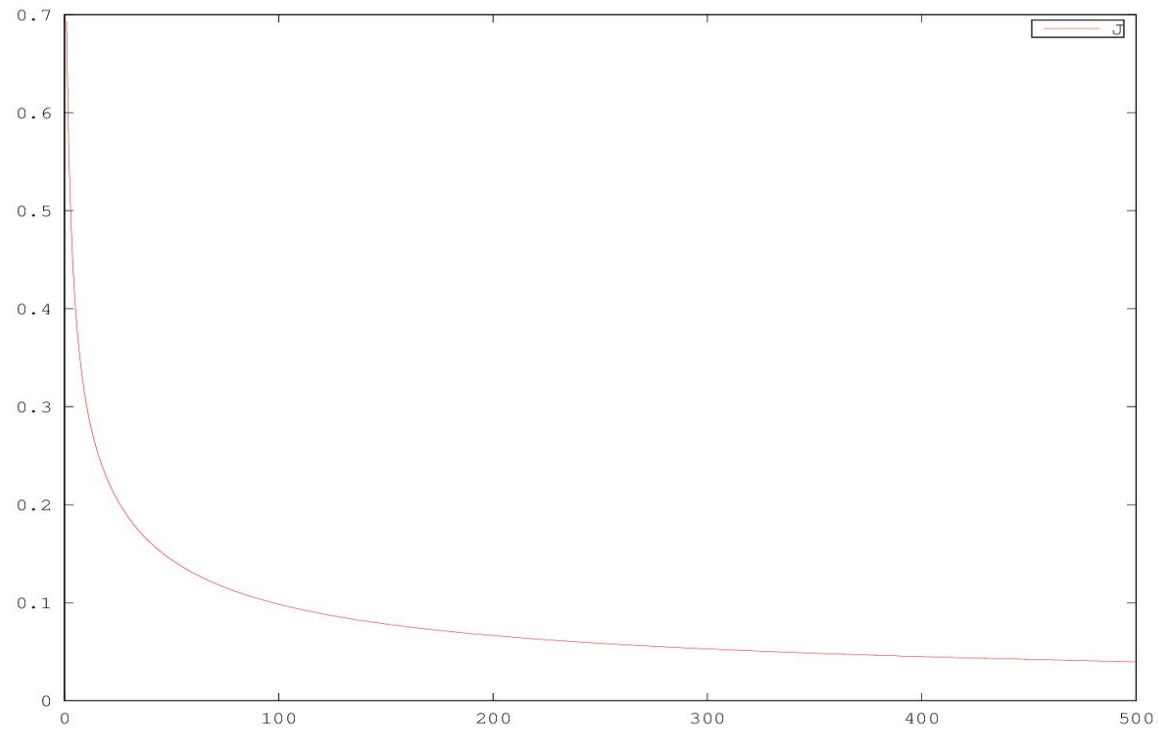
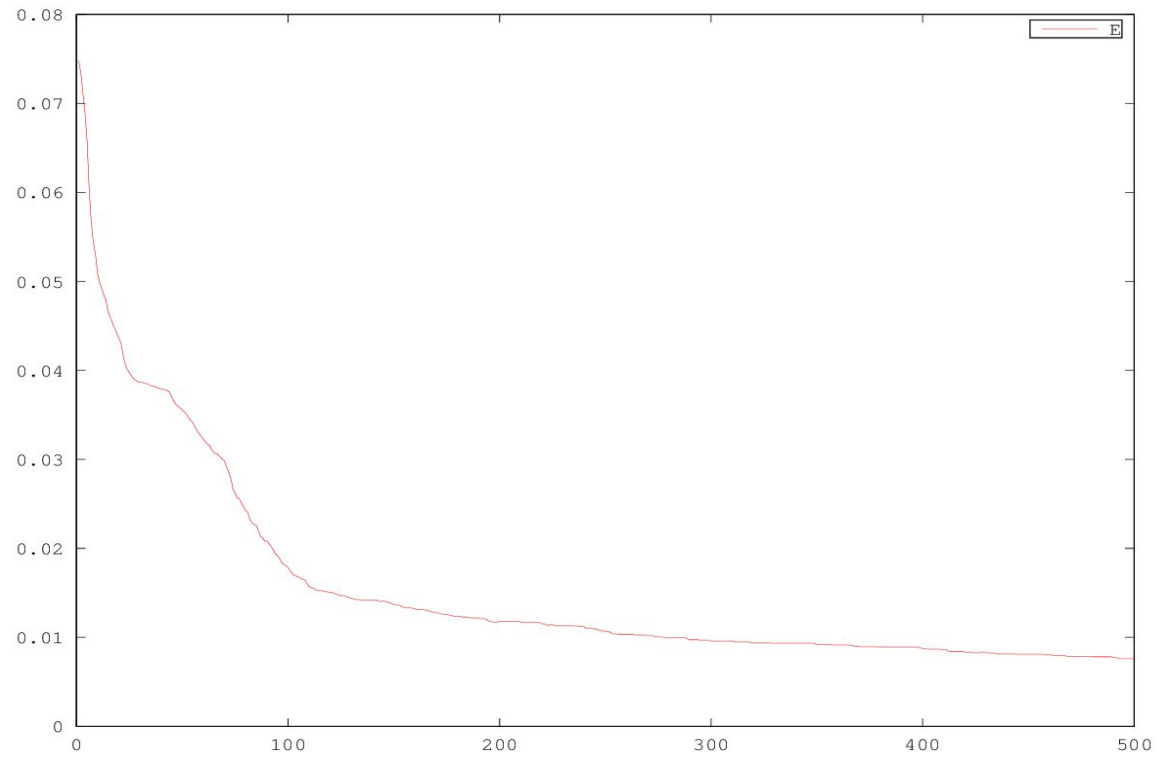


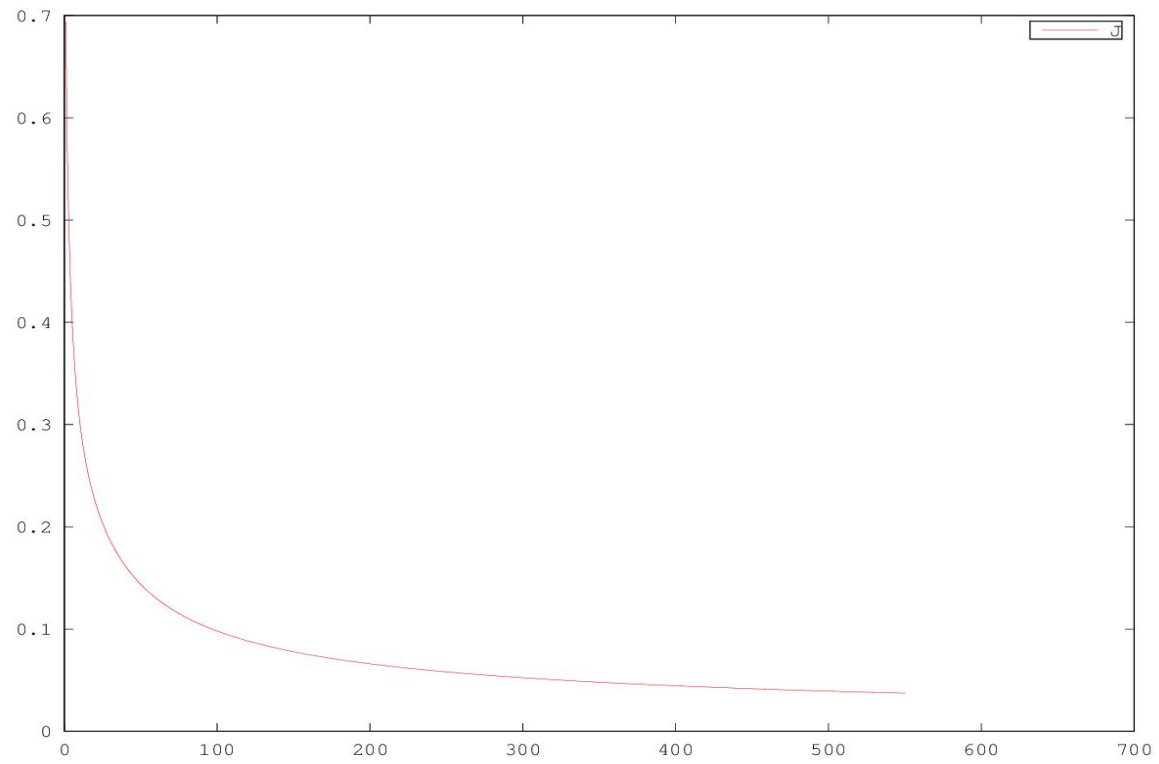
Imagen - Resultados algoritmo clasificador de emails

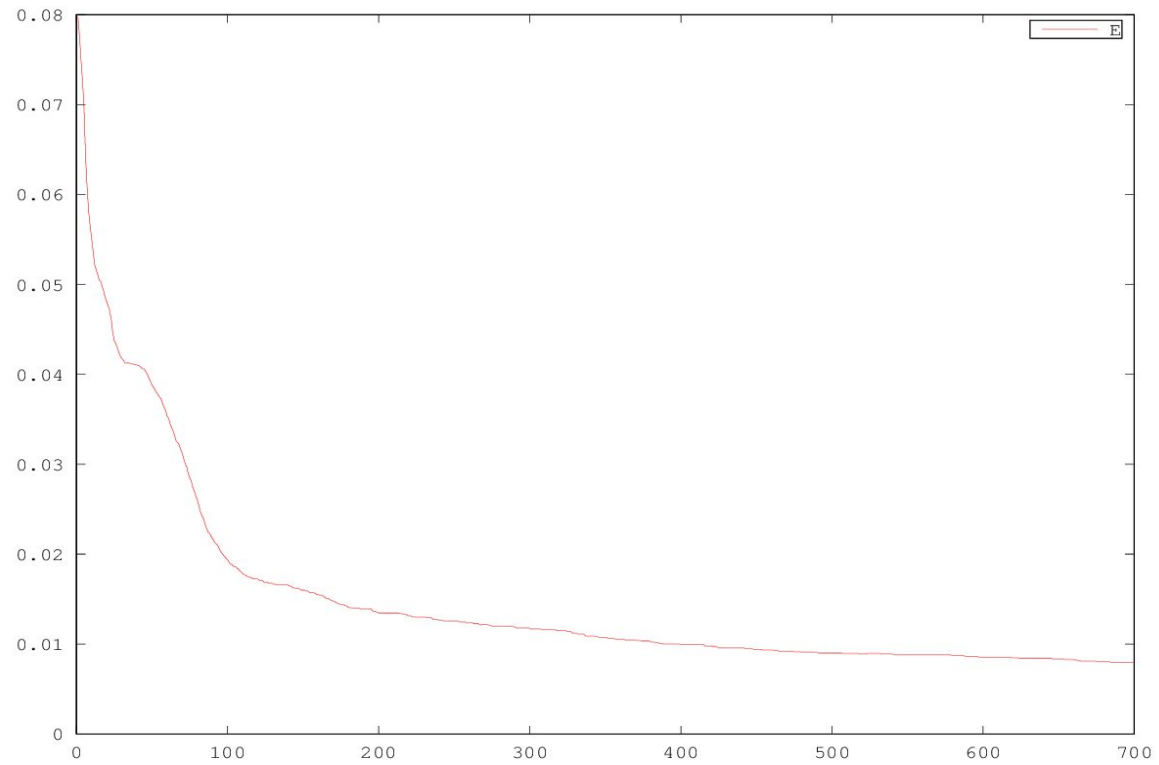
Tiempo (segundos)	Alpha	N# Iteraciones
732.997	0.01	500



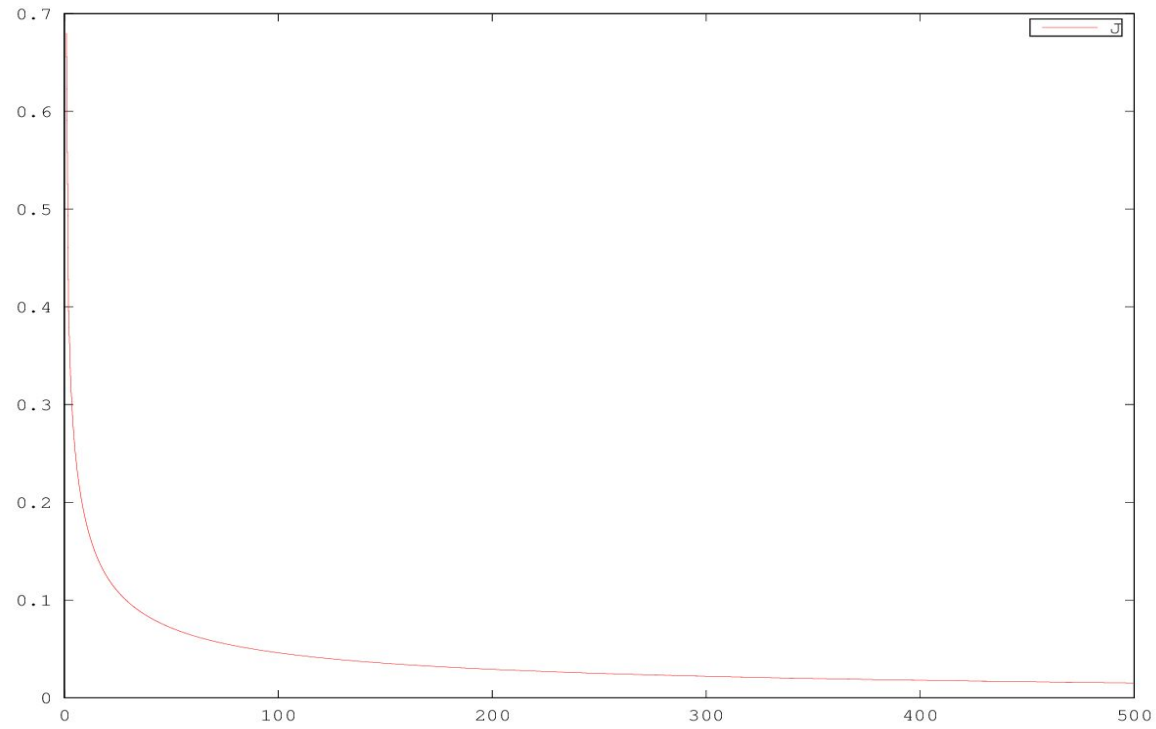


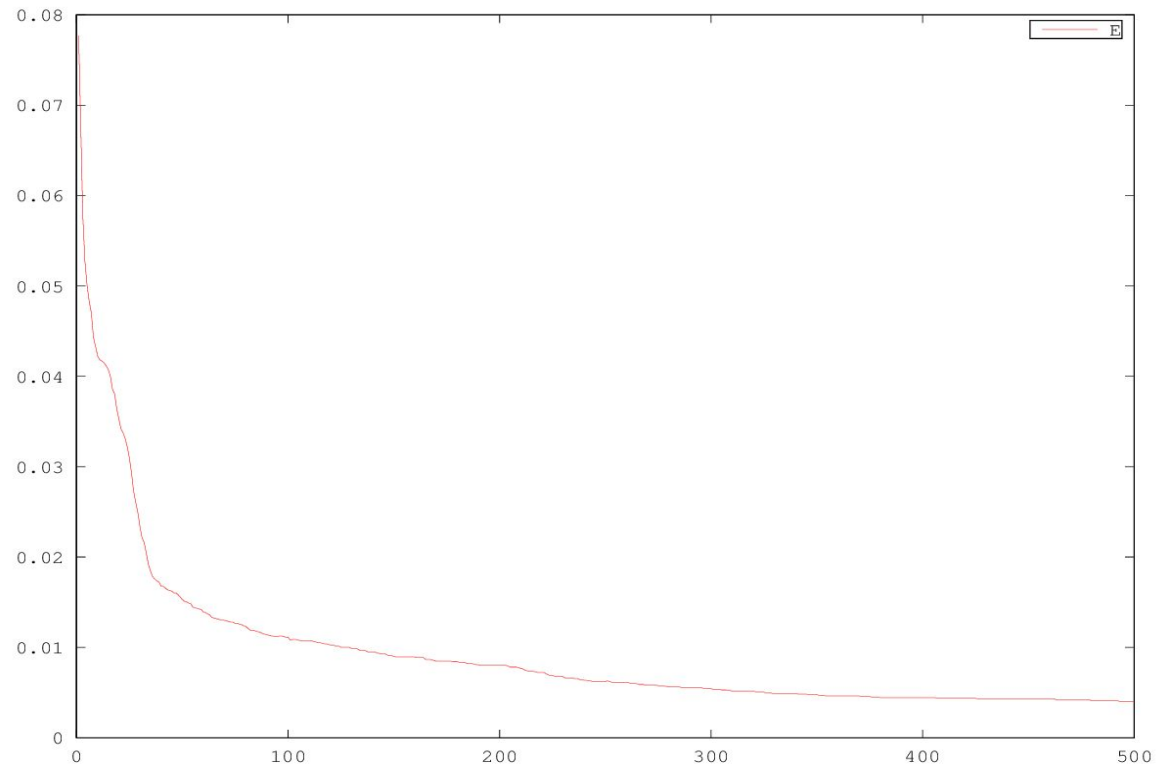
Tiempo (segundos)	Alpha	N# Iteraciones
1028.46	0.01	700





Tiempo (segundos)	Alpha - log(+0.01)	N# Iteraciones
732.169	0.03	500





Conclusiones :

1. El tiempo de ejecución es proporcional a las iteraciones y al tamaño del alpha.
2. El error disminuye a medida de que el dataset para entrenamiento aumenta.
3. El valor “num” que se suma en log(H+num) afecta directamente J, dependiendo de éste la gráfica no tiene valores para unos valores de iteraciones; “num” está también relacionado con la convergencia o no convergencia de la gráfica J para un determinado valor de alpha.

Entrega práctica parcial 1