

# PRÁCTICA 3

## MÉTODOS DE

### CLASIFICACIÓN



*Rubén García Mateos – Ingeniería del conocimiento*

## **Objetivo, Lenguaje utilizado y detalles de implementación.**

*El objetivo principal de esta práctica es implementar los diferentes métodos de clasificación vistos en clase (Bayes, Kmedias y Lloyd) en un determinado lenguaje de programación elegido por el alumno, donde el resultado final que se debe mostrar depende de cada algoritmo y será explicado a lo largo de esta memoria.*

*En mi caso, el lenguaje de programación que he utilizado es C++ junto con el IDE Visual Studio.*

*El algoritmo general consiste principalmente en cargar el archivo de texto con todos los datos que contiene de las muestras tomadas. El programa cargaría en diferentes estructuras y arrays los datos del archivo. Tenemos por un lado en la variable numMuestras el número de muestras que contiene el txt, en el vector datosMuestras tenemos todos los datos de las diferentes muestras, por ejemplo, de la muestra 1 almacenaría sus 4 coordenadas, de la muestra 2, sus otras 4 coordenadas y así sucesivamente, suponiendo que las muestras tienen 4 coordenadas, que en los diferentes ejemplos suministrados por mí (que se verán más adelante), hay casos en los que tenemos 2 coordenadas para cada muestra. En el vector clases, tenemos almacenado el nombre de las clases que contiene nuestro archivo, así como el número de clases que hay accediendo al tamaño de este array. Por último, en el programa utilizado tenemos dos variables de contador que indican el número de muestras que hay de la clase 1 y de la clase 2;*

*A partir de este momento es cuando se ejecutarían los diferentes algoritmos en base a la opción elegida por el usuario, ya que en la pantalla principal se nos muestra un menú para que seleccionemos el algoritmo que queremos utilizar.*

### **Algoritmo K-medias:**

*Este algoritmo ha sido implementado inicializando los centros facilitados por el profesor en el enunciado de la práctica.*

*Con estos centros ya inicializados, el programa pasaría a calcular la matriz de grados de pertenencia de la iteración en curso, esta matriz tendrá una dimensión de  $2 \times \text{NUMMUESTRAS}$ , donde en la primera columna se mostraría el grado de pertenencia de la muestra "i" a la clase 1, y en la segunda columna el grado de pertenencia de la muestra "i" a la clase 2. La columna 1 y 2 al mismo tiempo deberían sumar 1. Tras calcular esta matriz de grados de pertenencia, el programa pasaría a calcular los nuevos centros V1 y V2 asociados a esta matriz, y se comprobaría si los nuevos centros obtenidos y los anteriores tienen una distancia euclídea menor que un cierto exilon (límite) facilitado en el enunciado de la práctica. Si estas distancias son mayores que exilon se seguirá iterando y realizando el mismo proceso hasta conseguir una distancia menor, y cuando se consiga, los centros finales serán estos. Añadir como comentario que estas matrices de grados de pertenencia que se calcularán en el programa serán*

almacenadas en el archivo **MatrizGradosP.txt** que se generará cuando el usuario seleccione la opción de utilizar el algoritmo k-medias.

### **Algoritmo de Lloyd**

Este algoritmo ha sido implementado también inicializando los centros facilitados por el profesor en el enunciado de la práctica.

Con estos centros ya inicializados, el programa pasaría a calcular las distancias euclídeas que tenemos entre cada muestra “i” y el centro V1 y V2, seleccionaría la menor de las distancias, y calcularía el nuevo centro asociado a la menor de las distancias. Todo esto se haría para cada muestra y cuando se llegase a la última muestra, se realizaría el mismo proceso que en k-medias, se calcularía la distancia euclídea que se tiene entre los centros iniciales y los nuevos obtenidos en la última iteración, y se tendría que comprobar que la distancia entre los centros es menor que un nivel de tolerancia establecido en el enunciado de la práctica. En el caso de Lloyd el nivel de tolerancia es 10e-10. Añadir que en este algoritmo ya que son muchísimos cálculos para cada iteración, el programa tarda algo más de lo esperado ya que en el caso de la clasificación de la flor “Iris”, son 100 muestras lo que se facilita en el archivo y el programa tiene que ejecutar y mostrar cálculos para 100 muestras en cada iteración. Como estos datos son mostrados por consola, no se pueden visualizar todas las iteraciones sino solo los cálculos de la última. En el ejemplo de Lloyd.txt que facilito, se puede comprobar todas las iteraciones y cálculos ya que son 10 muestras lo que contiene el archivo.

### **Algoritmo de parametrización de Bayes**

En esta parte no se necesita inicialización de tal manera que al seleccionar la opción de este algoritmo el programa calcula los centros m1 y m2 que son realmente las medias, y los llama V1 y V2 para seguir semejanza con los anteriores algoritmos.

Posteriormente cuando ya ha calculado los centros, el programa continúa calculando las matrices de covarianza asociadas a estos centros. Estas matrices de covarianza y los centros asociados se calculan siguiendo las siguientes fórmulas:

$$\hat{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad C = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mathbf{m}})(\mathbf{x}_i - \hat{\mathbf{m}})^t$$

## Partes Opcionales

En lo relativo a las partes opcionales, He intentado realizar toda la práctica completa y creo que he podido conseguirlo. Se han realizado los 3 algoritmos de clasificación vistos en clase y además el programa clasifica una muestra facilitada por el usuario en la opción del algoritmo de Lloyd.

## Manual de Usuario

Para poder ejecutar el programa hay que descomprimir el proyecto en la carpeta raíz de proyectos de V.S. Dentro de Visual Studio pincharíamos archivo->abrir proyecto y seleccionamos el proyecto que hemos guardado. (Dentro del proyecto se incluyen varios ficheros de texto llamados "Iris2Clases.txt", "lloyd.txt" y "febrero2014.txt".

Nota: El proyecto completo contiene varios ficheros de texto para que se pueda probar por parte del profesor más ejemplos añadidos por el alumno para comprobar que funciona para todos los archivos de texto insertados. Si se quiere realizar pruebas de los otros archivos hay que salir del programa y volver a ejecutarlo para que nos pida el nombre del archivo que queremos simular.

Los nombres de los archivos que hay que introducir dentro del programa cuando se inicia para ejecutar los distintos algoritmos se corresponden con los siguientes (Escribimos uno de ellos):

Archivo1: "Iris2Clases.txt"

Archivo2: "Lloyd.txt" (Se corresponde con el ejercicio de las transparencias T4)

Archivo3: "febrero2014.txt" (Se corresponde con el ejercicio de ese examen)

El usuario al iniciar el programa debe escribir el nombre del archivo que quiera simular.

**IMPORTANTE:** Para ejecutar los diferentes archivos que tiene el programa hay que cambiar una variable constante del programa en el archivo de código. Esta variable es llamada NUM\_COL y hay que darle los siguientes valores para cada archivo de texto que queramos probar:

Para el Archivo1: "Iris2Clases.txt" → La variable NUM\_COL debe valer 100

Para el Archivo2: "Lloyd.txt" → La variable NUM\_COL debe valer 10

Para el Archivo3: "febrero2014.txt" → La variable NUM\_COL debe valer 6

```
//-----CONSTANTES-----
const int MAX_FIL = 2;
const int MAX_COL = 100; // <-----CONSTANTE QUE HAY QUE MODIFICAR PARA LOS DISTINTOS EJEMPLOS.
const int MAX_CELDAS = 5;
const double razon_Aprendizaje = 0.1; //Razon de aprendizaje para Lloyd
const double toleranciaLloyd = pow(10, -10);
const double toleranciaMedias = 0.01;
//-----VARIABLES GLOBALES AL PROGRAMA-----
typedef double matrizGradosP[MAX_FIL][MAX_COL];
typedef vector<double> datosMuestras;
typedef vector<double> centros;
typedef vector<double> datosX;
typedef double covarianza[MAX_CELDAS][MAX_CELDAS];
```

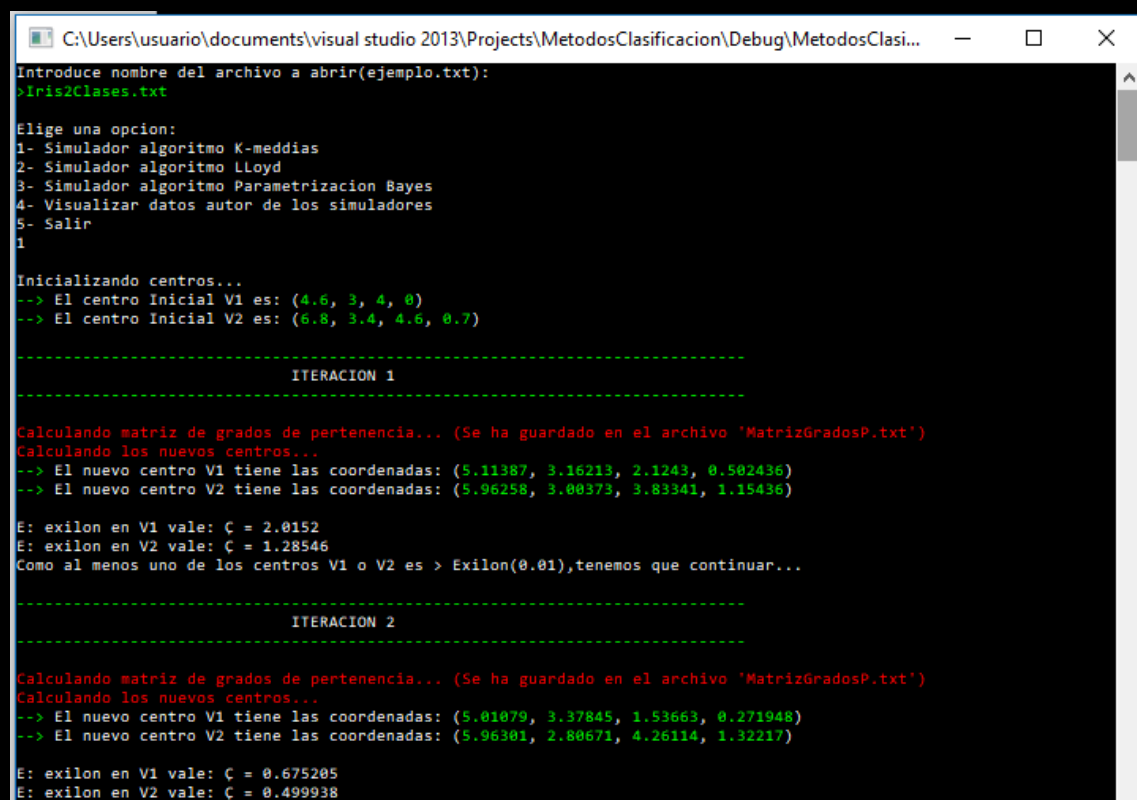
*\_Primero al iniciar el programa nos pediría el archivo de texto que queremos que se evalué para los diferentes algoritmos y una vez cargado, nos muestra el menú para que seleccionemos cual queremos simular:*

```
Introduce nombre del archivo a abrir(ejemplo.txt):
>Iris2Clases.txt

Elige una opcion:
1- Simulador algoritmo K-medias
2- Simulador algoritmo Lloyd
3- Simulador algoritmo Parametrizacion Bayes
4- Visualizar datos autor de los simuladores
5- Salir
-
```

### **Opción 1 K-medias**

*Al seleccionar la opción 1, el programa pasaría a simular el algoritmo k-medias con el archivo que le hemos introducido al comienzo del programa. El programa mostraría los centros iniciales que ha asignado, y comenzaría a mostrar todas las iteraciones que realiza. Como he mencionado antes, la matriz de grados de pertenencia de cada iteración se almacena en el directorio raíz del proyecto en el archivo "MatrizGradosP". Después se mostraría el valor de exilon que ha calculado entre los centros iniciales y los actuales calculados, y nos dice que como alguno de los centros es mayor que exilon, tenemos que continuar.*



```
C:\Users\usuario\documents\visual studio 2013\Projects\MetodosClasificacion\Debug\MetodosClasi...
Introduce nombre del archivo a abrir(ejemplo.txt):
>Iris2Clases.txt

Elige una opcion:
1- Simulador algoritmo K-medias
2- Simulador algoritmo Lloyd
3- Simulador algoritmo Parametrizacion Bayes
4- Visualizar datos autor de los simuladores
5- Salir
1

Iniciando centros...
--> El centro Inicial V1 es: (4.6, 3, 4, 0)
--> El centro Inicial V2 es: (6.8, 3.4, 4.6, 0.7)

-----
ITERACION 1
-----

Calculando matriz de grados de pertenencia... (Se ha guardado en el archivo 'MatrizGradosP.txt')
Calculando los nuevos centros...
--> El nuevo centro V1 tiene las coordenadas: (5.11387, 3.16213, 2.1243, 0.502436)
--> El nuevo centro V2 tiene las coordenadas: (5.96258, 3.00373, 3.83341, 1.15436)

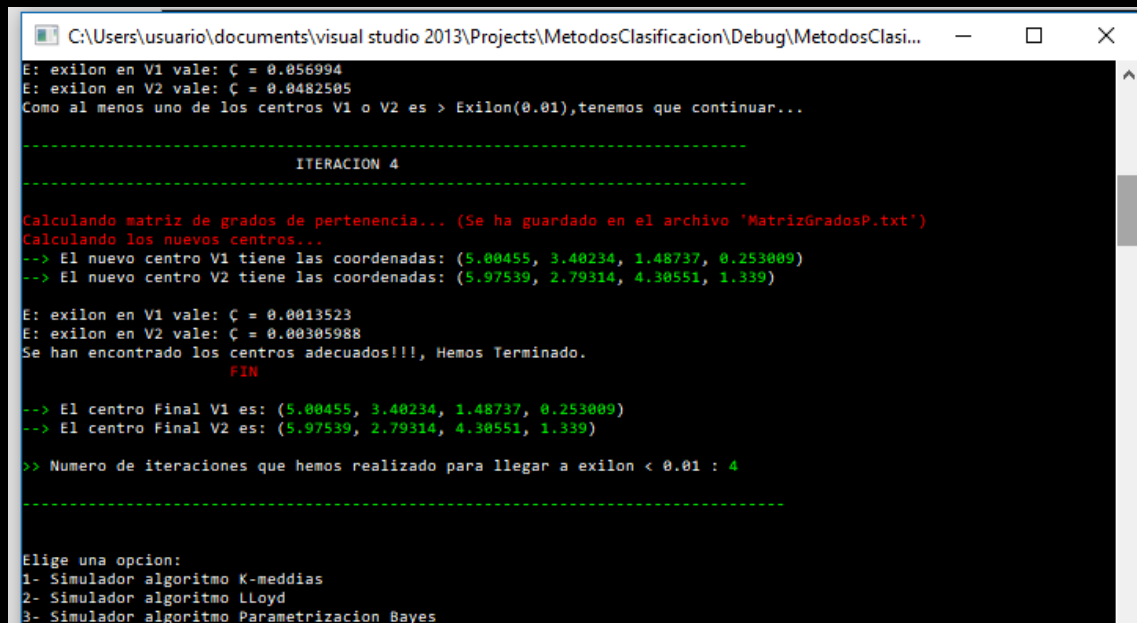
E: exilon en V1 vale:  $\epsilon = 2.0152$ 
E: exilon en V2 vale:  $\epsilon = 1.28546$ 
Como al menos uno de los centros V1 o V2 es > Exilon(0.01),tenemos que continuar...

-----
ITERACION 2
-----

Calculando matriz de grados de pertenencia... (Se ha guardado en el archivo 'MatrizGradosP.txt')
Calculando los nuevos centros...
--> El nuevo centro V1 tiene las coordenadas: (5.01079, 3.37845, 1.53663, 0.271948)
--> El nuevo centro V2 tiene las coordenadas: (5.96301, 2.80671, 4.26114, 1.32217)

E: exilon en V1 vale:  $\epsilon = 0.675205$ 
E: exilon en V2 vale:  $\epsilon = 0.499938$ 
```

Al terminar el algoritmo nos mostraría los cálculos de la última iteración junto con el centro final obtenido y los valores de exilon que podemos comprobar que ya son menores que la tolerancia que nos pide en la práctica que es 0.01. Y finalmente nos mostraría el número de iteraciones que se ha requerido hasta llegar al centro final y volvería a mostrar el menú por si queremos continuar probando mas algoritmos.



```
C:\Users\usuario\documents\visual studio 2013\Projects\MetodosClasificacion\Debug\MetodosClasi...
E: exilon en V1 vale: C = 0.056994
E: exilon en V2 vale: C = 0.0482505
Como al menos uno de los centros V1 o V2 es > Exilon(0.01),tenemos que continuar...

-----
ITERACION 4
-----

Calculando matriz de grados de pertenencia... (Se ha guardado en el archivo 'MatrizGradosP.txt')
Calculando los nuevos centros...
--> El nuevo centro V1 tiene las coordenadas: (5.00455, 3.40234, 1.48737, 0.253009)
--> El nuevo centro V2 tiene las coordenadas: (5.97539, 2.79314, 4.30551, 1.339)

E: exilon en V1 vale: C = 0.0013523
E: exilon en V2 vale: C = 0.00305988
Se han encontrado los centros adecuados!!!, Hemos Terminado.
FIN

--> El centro Final V1 es: (5.00455, 3.40234, 1.48737, 0.253009)
--> El centro Final V2 es: (5.97539, 2.79314, 4.30551, 1.339)

>> Numero de iteraciones que hemos realizado para llegar a exilon < 0.01 : 4

-----

Elige una opcion:
1- Simulador algoritmo K-medias
2- Simulador algoritmo Lloyd
3- Simulador algoritmo Parametrizacion Bayes
```

### **Opción 2: Algoritmo de Lloyd**

Al seleccionar la opción 2, el programa pasaría a simular el algoritmo de Lloyd con el archivo que le hemos introducido al comienzo del programa.

Como he comentado anteriormente, debido a la gran cantidad de cálculos que el programa debe realizar y que se muestran por consola, el programa tardará unos segundos en ejecutar todos estos cálculos que podremos comprobar posteriormente. Se calcularía las distancias entre los centros y cada una de las muestras del archivo y actualiza el centro en base a este cálculo e irá mostrando los centros que ha actualizado y como quedan en ese momento. Al llegar al final de las iteraciones nos mostraría como queda el centro final y el número de iteraciones que ha tenido que realizar para llegar a ese centro. En este algoritmo hemos limitado el número de iteraciones que debe de realizar el programa a 10, de tal manera que se ejecutará hasta obtener un exilon con una tolerancia determinada en el enunciado de la práctica que es  $10e-10$ , o con un número máximo de iteraciones. Para finalizar, El programa nos pedirá si queremos clasificar una muestra que le introduzcamos por teclado, que debe tener el mismo número de coordenadas que los centros, nos dirá a qué clase pertenece la muestra y nos volverá a mostrar el menú.

```
C:\Users\usuario\documents\visual studio 2013\Projects\MetodosClasificacion\Debug\MetodosClasi...
Como D1 es la mas pequena, actualizamos V1:
--> El nuevo centro V1 tiene las coordenadas: (5.08235, 3.47071, 1.47469, 0.227742)
--> El nuevo centro V2 tiene las coordenadas: (5.74403, 2.75161, 4.09664, 1.27468)

Como D1 es la mas pequena, actualizamos V1:
--> El nuevo centro V1 tiene las coordenadas: (5.12412, 3.47364, 1.45722, 0.224967)
--> El nuevo centro V2 tiene las coordenadas: (5.74403, 2.75161, 4.09664, 1.27468)

Como D1 es la mas pequena, actualizamos V1:
--> El nuevo centro V1 tiene las coordenadas: (5.10171, 3.43628, 1.4615, 0.212471)
--> El nuevo centro V2 tiene las coordenadas: (5.74403, 2.75161, 4.09664, 1.27468)

Como D1 es la mas pequena, actualizamos V1:
--> El nuevo centro V1 tiene las coordenadas: (5.03154, 3.39265, 1.44535, 0.211224)
--> El nuevo centro V2 tiene las coordenadas: (5.74403, 2.75161, 4.09664, 1.27468)

Como D1 es la mas pequena, actualizamos V1:
--> El nuevo centro V1 tiene las coordenadas: (5.03838, 3.39338, 1.45081, 0.210101)
--> El nuevo centro V2 tiene las coordenadas: (5.74403, 2.75161, 4.09664, 1.27468)
```

```
C:\Users\usuario\documents\visual studio 2013\Projects\MetodosClasificacion\Debug\MetodosClasi...
--> El nuevo centro V2 tiene las coordenadas: (5.82102, 2.77359, 4.21807, 1.29097)

Como D2 es la mas pequena, actualizamos V2:
--> El nuevo centro V1 tiene las coordenadas: (4.95783, 3.3773, 1.46849, 0.251329)
--> El nuevo centro V2 tiene las coordenadas: (5.74892, 2.74623, 4.09627, 1.27187)

Como D2 es la mas pequena, actualizamos V2:
--> El nuevo centro V1 tiene las coordenadas: (4.95783, 3.3773, 1.46849, 0.251329)
--> El nuevo centro V2 tiene las coordenadas: (5.74403, 2.75161, 4.09664, 1.27468)

>> Numero de iteraciones que hemos realizado para llegar a exilon < 0.01 : 6
--> El centro Final V1 es: (4.95783, 3.3773, 1.46849, 0.251329)
--> El centro Final V2 es: (5.74403, 2.75161, 4.09664, 1.27468)

-----
Quieres clasificar una muestra facilitada por ti? (s/n)
s
>> Introduce las coordenadas de la muestra que quieres clasificar
>> La muestra ha de tener 4 coordenadas:
2
2
2
2
--> La muestra introducida pertenece a la clase 1

Elige una opcion:
1- Simulador algoritmo K-meddias
2- Simulador algoritmo Lloyd
3- Simulador algoritmo Parametrizacion Bayes
4- Visualizar datos autor de los simuladores
```

### Opción 3: Algoritmo de parametrización de Bayes

Al seleccionar la opción 3, el programa pasaría a simular el algoritmo de Bayes con el archivo que le hemos introducido al comienzo del programa.

Como Bayes no necesita inicialización, el programa pasaría directamente a calcular a través de los datos procesados los nuevos centros M1 y M2 que en el programa se llaman V1 y V2 para mantener la semejanza con los otros algoritmos. Una vez haya calculado estos centros, pasaría a calcular las matrices de covarianza asociadas a estos centros y nos mostraría el resultado de estas matrices y de los centros calculados. Las matrices de covarianza se han calculado siguiendo la fórmula de las transparencias.

```
Elige una opcion:
1- Simulador algoritmo K-medias
2- Simulador algoritmo Lloyd
3- Simulador algoritmo Parametrizacion Bayes
4- Visualizar datos autor de los simuladores
5- Salir
3

Calculando Medias M1 y M2...
--> El nuevo centro V1 tiene las coordenadas: (5.006, 3.418, 1.464, 0.244)
--> El nuevo centro V2 tiene las coordenadas: (5.936, 2.77, 4.26, 1.326)

La matriz de covarianza C1:

| 0.12 0.098 0.016 0.01 |
| 0.098 0.14 0.011 0.011 |
| 0.016 0.011 0.03 0.0056 |
| 0.01 0.011 0.0056 0.011 |

La matriz de covarianza C2:

| 0.26 0.083 0.18 0.055 |
| 0.083 0.097 0.081 0.04 |
| 0.18 0.081 0.22 0.072 |
| 0.055 0.04 0.072 0.038 |

Elige una opcion:
1- Simulador algoritmo K-medias
2- Simulador algoritmo Lloyd
3- Simulador algoritmo Parametrizacion Bayes
4- Visualizar datos autor de los simuladores
```

### Opciones 4 y 5

La opción 4 del programa nos mostraría mis datos propiamente escritos, la asignatura que se está cursando y la práctica elaborada.

Con la opción 5 saldremos del programa.



## Comentarios Adicionales a la práctica

Como comentario adicional añadir que también se puede cambiar la tolerancia de cada algoritmo a través del código. Con solo cambiar los datos de las variables `toleranciaLloyd` y `toleranciakMeddias` el programa se ejecutaría con estos datos.

```
//-----CONSTANTES-----  
const int MAX_FIL = 2;  
const int MAX_COL = 100; // <-----CONSTANTE QUE HAY QUE MODIFICAR PARA LOS DISTINTOS EJEMPLOS.  
const int MAX_CELDAS = 5;  
const double razon_Aprendizaje = 0.1; //Razon de aprendizaje para Lloyd  
const double toleranciaLloyd = pow(10, -10);  
const double toleranciakMeddias = 0.01;  
//-----
```

La práctica me ha parecido bastante entretenida pero compleja de implementar en programación ya que las fórmulas que se utilizan de las transparencias utilizan muchas llamadas al mismo método y puede haber valores inesperados debido a la gran cantidad de cálculos que deben ejecutar los algoritmos. He tenido que dedicarle bastantes horas para implementar todo, pero me ha costado especialmente más el algoritmo k-medias y Bayes (Matrices de covarianza).

Pienso que debido a la dificultad de la práctica y el tiempo que he tenido que emplear en ella junto con los detalles de implementación realizados y que está implementada la parte de clasificación de una muestra junto con los 3 algoritmos, mi nota debería estar en torno al (9.5-10.0).

Rubén García Mateos, 02551077D, Ingeniería del conocimiento. 4E