Prácticas

Introducción a la Minería de Datos

Grado en Ingeniería Informática

Escuela Politécnica Superior de Córdoba

Universidad de Córdoba.

4ª Curso, Especialidad de Computación

Enrique Álvarez Gallego (i32@algae.es)
45742699T

Índice

PRÁCTICA 1. Preprocesamiento y exploración de datos

Ejercicio 1

Usando como clasificador el método J48 compruebe el efecto de los dos filtros estudiados sobre el error de clasificación en uno de los conjuntos de datos de la UCI MLR.

El conjunto de datos utilizados es iris.arff

Primero clasificaremos con el método J48 el conjunto de datos, sin aplicarle ningún filtro. A continuación, clasificaremos el conjunto de datos con dicho método, pero habiendo filtrado los mismos primero con alguno de los siguientes filtros:

- RandomProjection: reduce las dimensiones de los datos, proyectándolos en un subespacio de menor dimensión, utilizando una matriz aleatoria con columnas de longitud unitaria. Para ello, aplica primero el filtro NominalToBinary para convertir los atributos en numéricos antes de reducir la dimensión.
- RemoveUseless: elimina los atributos que no varían en absoluto, o que varían demasiado.

Una vez hecho, se pasará a repetir el proceso pero con el filtro no utilizado anteriormente.

iris	Instancias mal clasificadas (%)	Root mean squared error
Sin filtro	4	0,1586
RandomProjection	2,6667	0,1322
RemoveUseless	4	0,1586

Se puede comprobar que aplicando **RandomProjection** se obtienen mejores resultados, al menos con las medidas de error que se han considerado, lo cuál tiene sentido, ya que como se ha explicado antes, RemoveUseless nos elimina los atributos que no sirven, mientras que RandomProjection reduce la dimensión de los datos, haciendo que sea más fácil clasificarlos.

Ejercicio 2

Usando como clasificador el método J48 compruebe el efecto de la normalización, la estandarización y el análisis en componentes principales sobre el error de clasificación en los conjuntos de datos.

Trabajaremos sobre el mismo conjunto de datos que el ejercicio anterior. Además, utilizaremos otras dos bases de datos (diabetes.arff y weather.arff).

Sabiendo lo bien y mal que clasifica el J48 sobre la base de datos sin filtros (ejercicio anterior), vamos a comprobar como clasifica aplicándole 3 filtros:

- Normalize
- Standarize
- PrincipalComponents

iris	Instancias mal clasificadas (%)	Root mean squared error
Sin filtro	4	0,1586
Normalize	4	0,1586
Standarize	4	0,1586
PrincipalComponents	15,34	0,2906

La aplicación de los diferentes filtros no altera las diferentes medidas de errores, menos en el caso de Principal Componentes, con la que empeora considerablemente el porcentaje de patrones mal clasificados.

diabetes	Instancias mal clasificadas (%)	Root mean squared error
Sin filtro	26,1719	0,4463
Normalize	26,1719	0,4463
Standarize	26,1719	0,4463
PrincipalComponents	29,2969	0,4543

La aplicación de los diferentes filtros no altera las diferentes medidas de errores, menos en el caso de Principal Componentes, con la que empeora considerablemente el porcentaje de patrones mal clasificados.

weather	Instancias mal clasificadas (%)	Root mean squared error
Sin filtro	35,7143	0,4818
Normalize	35,7143	0,4818
Standarize	35,7143	0,4818
PrincipalComponents	35,7143	0,5804

La aplicación de los diferentes filtros no altera las diferentes medidas de erroes, menos en el caso de Principal Componentes, con la que empeora considerablemente el porcentaje de patrones mal clasificados.

A diferencia que para los otros dos conjuntos de datos utilizados, en este caso el porcentaje de instancias mal clasificadas no aumenta. Esto se debe a que el número de instancias en este conjunto de datos es muy pequeño (14 para ser exactos), por lo que no ha variado a pesar del filtro. Sin embargo, RMSE si que ha empeorado, manteniendo la dinámica de los otros dos conjuntos de datos.

PRÁCTICA 2. Clasificación y evaluación de modelos

Ejercicio 1

Seleccione 3 clasificadores dentro de los disponibles de weka y 5 conjuntos de datos. No use combinaciones de modelos.

Se van a utilizar las siguientes bases de datos:

- Diabetes
- Iris
- Weather
- Supermarket
- Vote

Los clasificadores que se van a utilizar son:

- IB1
- RBFNetwork
- NaiveBayes

Ejercicio 2

Use como método para obtener el error la validación cruzada de 10 particiones. Ejecute para cada clasificador seleccionado un entrenamiento y anote el error.

Diabetes	Instancias mal clasificadas (%)	Root mean squared error
IB1	29,8177 (3)	0,5461
RBFNetwork	24,6094 (2)	04191
NaiveBayes	23,6979 (1)	0,4168

Iris	Instancias mal clasificadas (%)	Root mean squared error
IB1	4,6667 (2,5)	0,1764
RBFNetwork	4,6667 (2,5)	0,1585
NaiveBayes	4 (1)	0,155

Weather	Instancias mal clasificadas (%)	Root mean squared error
IB1	21,4286 (1)	0,4629
RBFNetwork	57,1429 (3)	0,7559
NaiveBayes	35,7143 (2)	0,543

Supermarket	Instancias mal clasificadas (%)	Root mean squared error
IB1	62,1569 (3)	0,7884
RBFNetwork	36,287 (1,5)	0,4808
NaiveBayes	36,287 (1,5)	0,4808

Vote	Instancias mal clasificadas (%)	Root mean squared error
IB1	7,5862 (2)	0,2754
RBFNetwork	5,5172 (1)	0,2192
NaiveBayes	9,8851 (3)	0,2977

Marcamos para la cada base de datos, el orden de qué algoritmos han obtenido mejor resultado, siendo para el mejor el valor de 1, y para el peor de 3.

Ahora calcularemos el rango medio para cada algoritmo, teniendo en cuenta los valores asignados en cada base de datos.

	IB1	RBFNetwork	NaiveBayes
Rango medio	1,7	2	1,7

Una vez se tiene el rango medio de los algoritmos, se debe comprobar que la diferencia entre ellos (mirando a pares) nunca supera la diferencia crítica (que la calcularemos en el ejercicio 4 con el test de Nememyi).

Ejercicio 3

Use el test de Wilcoxon de comparación de dos algoritmos sobre N problemas y aplíquelo a dos de los algoritmos anteriores. Obtenga el rango de Friedman para cada clasificador y configuración y represente gráficamente los resultados. Aplique el test de Iman-Davenport sobre los tres clasificadores.

Para aplicar el test de **Wilcoxon** se han cogido los algoritmos de dos en dos:

- IB1 ↔ RBFNetwork

  V = 55, p-value = 0.005889

Puesto que el p-value obtenido es menor al nivel de confianza (0.05), se rechaza la hipótesis nula, por lo que con un 95% de confianza se puede decir que los dos algoritmos son distintos.

- IB1 ↔ NaiveBayes

  V = 55, p-value = 0.001953

Puesto que el p-value obtenido es menor al nivel de confianza (0.05), se rechaza la hipótesis nula, por lo que con un 95% de confianza se puede decir que los dos algoritmos son distintos.

- RBFNetwork ↔ NaiveBayes
  V = 55, p-value = 0.005889

Puesto que el p-value obtenido es menor al nivel de confianza (0.05), se rechaza la hipótesis nula, por lo que con un 95% de confianza se puede decir que los dos algoritmos son distintos.

Para aplicar el rango de **Friedman**, se han cogido los tres algoritmos en conjunto:

- IB1 ↔ NaiveBayes ↔ RBFNetwork: utiliza la mediana de los datos (21.4286, 23.6979, 24.6094)

  Friedman chi-squared = 1, df = 2, p-value = 0.6065

Puesto que el p-value obtenido es mayor al nivel de confianza (0.05), no se rechaza la hipótesis nula, por lo que no se puede decir que los algoritmos sea distintos.

Para aplicar el test de **Iman-Davenport** se han cogido los tres algoritmos en conjunto:

- IB1 ↔ NaiveBayes ↔ RBFNetwork:

  Corrected Friedman's chi-squared = 0.3956, df1 = 2, df2 = 8, p-value = 0.6857

Puesto que el p-value obtenido es mayor al nivel de confianza (0.05), no se rechaza la hipótesis nula, por lo que no se puede decir que los algoritmos sea distintos.

Ejercicio 4

Use el test de Nememyi para comparar todos los 3 métodos.

Para aplicar el test de **Nememyi** se han cogido los tres algoritmos en conjunto:

- IB1 ↔ NaiveBayes ↔ RBFNetwork:

  Critical difference = 1.6873, k = 3, df = 12

Si el rango medio es mayor a la diferencia critica, los algoritmos serán distintos.

Ejercicio 5

Enuncie las conclusiones del estudio.

Como he mencionado en el ejercicio 2, en caos de que las diferencias entre los rangos medios sean mayores a la diferencia crítica calculada con el test de Nememyi, esos algoritmos serían distintos.

- RBFNetwork ↔ IB1:

  2 - 1,7 = 0,3 → 0,3 < 1,68 → Los algoritmos no tiene una diferencia significatica.

- RBFNetwork ↔ NaiveBayes:

  2 - 1,7 = 0,3 → 0,3 < 1,68 → Los algoritmos no tiene una diferencia significatica.

- NaiveBayes ↔ IB1:

  1,7 - 1,7 = 0 → 0 < 1,68 → Los algoritmos no tiene una diferencia significatica.

Excepto para el test de Wilcoxon (que se realiza con los algoritmos de dos en dos), para el resto de variables calculadas, nos indica que no hay diferencias significativas entre los algoritmos elegidos, al menos con las bases de datos trabajadas.

Puesto que no hay diferencias significativas, sería interesante utilizar aquel que utilice menos recursos de la máquina y aquel que tarde menos tiempo.

PRÁCTICA 3. Clasificación avanzada

Ejercicio 1

Seleccione un algoritmo de clasificación de los disponibles en Weka y al menos 10 conjuntos de datos.

Se van a utilizar las siguientes bases de datos:

- Diabetes
- Iris
- Weather (o el nominal)
- Supermarket
- Vote
- Contact-lenses
- Glass
- Ionosphere
- Labor
- Soybean

Los clasificadores que se van a utilizar son: (Solo hacia falta 1, no 3. Para el siguiente ejercicio, no consigo resultados para RBFNetwork , por lo que si al final hago solo 1, mejor quedarme con IB1 o NaiveBayes) (tambien es verdad, que para hacer el apartado opcional, es necesario usar un clasificador distinto, asique mejor quedarse los dos).

- IB1
- RBFNetwork
- NaiveBayes

Ejercicio 2

Aplique el método base a cada uno de los conjuntos y anote los resultados obtenidos.

Diabetes	Instancias mal clasificadas (%)	Root mean squared error
IB1	29,8177	0,5461
RBFNetwork	24,6094	04191
NaiveBayes	23,6979	0,4168

Iris	Instancias mal clasificadas (%)	Root mean squared error
IB1	4,6667	0,1764
RBFNetwork	4,6667	0,1585
NaiveBayes	4	0,155

Weather	Instancias mal clasificadas (%)	Root mean squared error
IB1	21,4286	0,4629
RBFNetwork	57,1429	0,7559
NaiveBayes	35,7143	0,543

Supermarket	Instancias mal clasificadas (%)	Root mean squared error
IB1	62,1569	0,7884
RBFNetwork	36,287	0,4808
NaiveBayes	36,287	0,4808

Vote	Instancias mal clasificadas (%)	Root mean squared error
IB1	7,5862	0,2754
RBFNetwork	5,5172	0,2192
NaiveBayes	9,8851	0,2977

Contact-lenses	Instancias mal clasificadas (%)	Root mean squared error
IB1	37,5	0,5
RBFNetwork	16,667	0,3349
NaiveBayes	29,1667	0,3326

Glass	Instancias mal clasificadas (%)	Root mean squared error
IB1	29,4393	0,29
RBFNetwork	36,4486	0,2813
NaiveBayes	51,4019	0,3399

Ionosphere	Instancias mal clasificadas (%)	Root mean squared error
IB1	13,6752	0,3698
RBFNetwork	7,6923	0,2584
NaiveBayes	17,3789	0,3935

Labor	Instancias mal clasificadas (%)	Root mean squared error
IB1	17,5439	0,4189
RBFNetwork	5,2632	0,2322
NaiveBayes	10,5263	0,2637

Soybean	Instancias mal clasificadas (%)	Root mean squared error
IB1	10,1025	0,1031
RBFNetwork	9,6633	0,0934
NaiveBayes	7,0278	0,0817

Ejercicio 3

Aplique el método de combinación de clasificadores Bagging a cada uno de los conjuntos y anote los resultados obtenidos.

Diabetes	Instancias mal clasificadas (%)	Root mean squared error
IB1	29,8177	0,4826
RBFNetwork	24,0885	0,4093
NaiveBayes	23,0469	0,416

Iris	Instancias mal clasificadas (%)	Root mean squared error
IB1	4,6667	0,1562
RBFNetwork	4,6667	0,1537
NaiveBayes	3,3334	0,1491

Weather	Instancias mal clasificadas (%)	Root mean squared error
IB1	28,5714	0,4855
RBFNetwork	57,1429	0,6242
NaiveBayes	50	0,5708

Supermarket	Instancias mal clasificadas (%)	Root mean squared error
IB1	62,7188	0,78
RBFNetwork	36,287	0,4808
NaiveBayes	36,287	0,4808

Vote	Instancias mal clasificadas (%)	Root mean squared error
IB1	8,046	0,2468
RBFNetwork	5,977	0,2163
NaiveBayes	10,1149	0,2966

Contact-lenses	Instancias mal clasificadas (%)	Root mean squared error
IB1	29,1667	0,3563
RBFNetwork	37,5	0,3755
NaiveBayes	29,1667	0,3419

Glass	Instancias mal clasificadas (%)	Root mean squared error
IB1	30,8411	0,2568
RBFNetwork	28,972	0,2445
NaiveBayes	48,1308	0,3085

Ionosphere	Instancias mal clasificadas (%)	Root mean squared error
IB1	12,8205	0,3434
RBFNetwork	7,9772	0,2569
NaiveBayes	17,094	0,3641

Labor	Instancias mal clasificadas (%)	Root mean squared error
IB1	14,0351	0,3258
RBFNetwork	10,5263	0,2922
NaiveBayes	10,5263	0,2319

Soybean	Instancias mal clasificadas (%)	Root mean squared error
IB1	8,4919	0,0838
RBFNetwork	7,0278	0,0817
NaiveBayes	7,0278	0,0817

Ejercicio 4

Seleccione al menos dos algoritmos de Boosting y aplique estos algoritmos a cada uno de los conjuntos y anote los resultados obtenidos.

Diabetes	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	26,0417	0,4186
RandomForest	26,5625	0,4319

Iris	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	6,6667	0,1953
RandomForest	6	0,1977

Weather	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	57,1429	0,6896
RandomForest	50	0,5578

Supermarket	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	25,0486	0,4134
RandomForest	36,287	0,4808

Vote	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	4,8276	0,191
RandomForest	5,5172	0,2006

Contact-lenses	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	33,3334	0,4181
RandomForest	33,3334	0,3343

Glass	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	56,5421	0,3049
RandomForest	31,3084	0,2492

Ionosphere	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	13,6752	0,3153
RandomForest	9,6866	0,2699

Labor	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	15,7895	0,3971
RandomForest	14,0351	0,3425

Soybean	Instancias mal clasificadas (%)	Root mean squared error
AdaBoostM1	72,0351	0,2032
RandomForest	13,0307	0,1109

Ejercicio 5

Compare si hay diferencias significativas entre ellos usando el test de Iman-Davenport. Si es así, aplique el procedimiento de Wilcoxon para comparar cada método de agrupación con el clasificador base.

Para aplicar el test de **Iman-Davenport** se han cogido los tres algoritmos en conjunto:

- AdaBoostM1 ↔ RandomForest:

Corrected Friedman's chi-squared = 0.89011, df1 = 1, df2 = 9, p-value = 0.3701

Puesto que el p-value obtenido es mayor al nivel de confianza (0.05), no se rechaza la hipótesis nula, por lo que no se puede decir que los algoritmos sea distintos.

Como no existe una diferencia significativa entre ellos, no aplicamos el procedimiento de Wilcoxon.

Ejercicio 7

Enuncie las conclusiones del estudio

No hay diferencia entre los dos algoritmos de boosting escogidos.

Puesto que no hay diferencias significativas, sería interesante utilizar aquel que utilice menos recursos de la máquina y aquel que tarde menos tiempo.

Práctica 4: Clasificación usando método multiclase

Ejercicio 1

Seleccione un algoritmo clasificación de los disponibles en Weka que sea capaz de resolver problemas de más de dos clases y al menos 10 conjuntos de datos de más de 2 clases.

Se van a utilizar las siguientes bases de datos:

- Iris
- Contact-lenses
- Glass
- Zoo
- vehicle
- nursey
- flags
- dermatology
- car
- lymph

Algoritmos multiclase:

- trees.J48

Ejercicio 2

Aplique el clasificador base a cada uno de los conjuntos y anote los resultados obtenidos.

	Instancias mal clasificadas (%)	Root mean squared error
Iris	4	0,1586
Contact-lenses	16,6667	0,3249
Glass	33,1776	0,2897
Zoo	7,9208	0,14
Vehicle	27,5414	0,3355
Nursey	2,9475	0,0951
Flags	40,7216	0,2782
Dermatology	6,0109	0,1365
Car	7,6389	0,1718
Lymph	22,973	0,3151

Ejercicio 3

Aplique los métodos multiclase ovo, ova y ecoc a cada uno de los conjuntos de datos y anote los resultados obtenidos.

Iris	Instancias mal clasificadas (%)	Root mean squared error
One vs One	22,973	0,3459
One vs All	19,5946	0,3792
Exhaustive correction code	20,9459	0,3683

Contact-lenses	Instancias mal clasificadas (%)	Root mean squared error
One vs One	29,1667	0,4082
One vs All	29,1667	0,378
Exhaustive correction code	29,1667	0,378

Glass	Instancias mal clasificadas (%)	Root mean squared error
One vs One	28,0374	0,3117
One vs All	29,9065	0,339
Exhaustive correction code	24,7664	0,3245

Zoo	Instancias mal clasificadas (%)	Root mean squared error
One vs One	8,9109	0,3032
One vs All	7,9208	0,3336
Exhaustive correction code	3,9604	0,3115

Vehicle	Instancias mal clasificadas (%)	Root mean squared error
One vs One	28,7234	0,3537
One vs All	30,8511	0,3867
Exhaustive correction code	27,1868	0,3747

Nursey	Instancias mal clasificadas (%)	Root mean squared error
One vs One	2,9012	0,3181
One vs All	2,9012	0,3555
Exhaustive correction code	2,909	0,3298

Flags	Instancias mal clasificadas (%)	Root mean squared error
One vs One	41,2371	0,3016
One vs All	40,2062	0,3248
Exhaustive correction code	41,2371	0,3165

Dermatology	Instancias mal clasificadas (%)	Root mean squared error
One vs One	3,5519	0,3117
One vs All	8,4699	0,3485
Exhaustive correction code	3,0055	0,3242

Car	Instancias mal clasificadas (%)	Root mean squared error
One vs One	5,9028	0,3201
One vs All	7,9861	0,3606
Exhaustive correction code	8,2176	0,343

Lymth	Instancias mal clasificadas (%)	Root mean squared error
One vs One	22,973	0,3459
One vs All	19,5946	0,3792
Exhaustive correction code	20,9459	0,3683

Ejercicio 4

Compare si hay diferencias significativas entre ellos usando el test de Iman-Davenport. Si es así, aplique el procedimiento de Wilcoxon para comparar cada método multiclase con el clasificador base y los diferentes métodos entre ellos.

Para aplicar el test de **Iman-Davenport** se han cogido los tres algoritmos en conjunto:

- OvO ↔ OvA ↔ ECOC:

Corrected Friedman's chi-squared = 0.30233, df1 = 2, df2 = 18, p-value = 0.7428

Puesto que el p-value obtenido es mayor al nivel de confianza (0.05), no se rechaza la hipótesis nula, por lo que no se puede decir que los algoritmos sea distintos.

Como no existe una diferencia significativa entre ellos, no aplicamos el procedimiento de Wilcoxon.

Ejercicio 5

Enuncie las conclusiones del estudio

No hay diferencia entre los dos algoritmos de multiclase probados.

Puesto que no hay diferencias significativas, sería interesante utilizar aquel que utilice menos recursos de la máquina y aquel que tarde menos tiempo.

Práctica 5: Reglas de asociación

Ejercicio 1

Seleccione un conjunto de datos de los suministrados con el paquete Weka. Para que se pueda usar el método a priori es necesario que los conjuntos no contengan variables numéricas.

Se ha seleccionado la base de datos wehater.nominal.arff.

Ejercicio 2

Usando la herramienta Weka y el método a priori estudie el efecto del umbral de soporte mínimo en el número de itemsets seleccionados.

La variable que controla el umbral de soporte mínimo en el número de itemsets seleccionados es lowerBoundMinSupport.

lowerBoundMinSupport	Size L(1)	Size L(2)	Size L(3)	Size L(4)	Reglas
0,1	12	47	39	6	10
0,2	12	26	4	0	8
0,3	12	9	1	0	3
0,4	6	2	0	0	0
0,5	0	0	0	0	0

Se puede comprobar que conforme baja el umbral de soporte de números de item seleccionados, baja el número de reglas creado. Interesa que el umbral sea lo más alto posible, ya que eso significaría que los items seleccionados tienen una gran representación en el conjunto de datos. Pero también se necesita que se generen reglas, por lo que, en este caso, el umbral máximo será 0,3.

Ejercicio 3

Usando el soporte mínimo que considere más apropiado según los resultados del ejercicio anterior realice un estudio del efecto del umbral mínimo de confianza en el número de reglas seleccionadas.

La variable que controla la confianza es MinMetric. Se va a probar diferentes valores, tanto para con un umbral de 0,2 como para un umbral de 0,3.

Para un umbral de soporte mínimo de 0.2:

MinMetric (Confidence)	Size L(1)	Size L(2)	Size L(3)	Size L(4)	Reglas
0,9	12	26	4	0	8
0,8	12	26	4	0	10
0,7	12	26	4	0	17
0,6	12	26	4	0	35
0,5	12	26	4	0	54
0,4	12	26	4	0	65

Para un umbral de soporte mínimo de 0.3:

MinMetric (Confidence)	Size L(1)	Size L(2)	Size L(3)	Size L(4)	Reglas
0,9	12	9	1	0	3
0,8	12	9	1	0	5
0,7	12	9	1	0	6
0,6	12	9	1	0	12
0,5	12	9	1	0	21
0,4	12	9	1	0	24

A medida que se va disminuyendo la confianza, va aumentando el número de reglas. Pero, aun que para valores más bajos de confianza se creen mayor número de reglas, estas tendran una confianza menor, es decir, serán ciertas en un porcentaje inferior. Lo que interesa es que las reglas sean lo más ciertas posibles, por lo que nos quedaremos con las reglas creadas para una confianza alto.

En este caso en concreto, se obtienen mejores resultados con un umbral de 0,2, y se puesto que el número de reglas de diferencia entre una confianza de 0,9 y 0,8, nos quedaremos con las 8 reglas creadas para una confianza de 0,9.

Práctica 6: Agrupación y evaluación de resultados

Ejercicio 1

Seleccione al menos cinco problemas de los disponibles en el paquete de Weka o en el repositorio de la UCI como ejemplos. Use problemas que solo contengan atributos numéricos.

Se van a utilizar las siguientes bases de datos:

- Diabetes
- Iris
- Glass
- Ionosphere
- Segment-challenge

Ejercicio 2

Implemente el método de agrupación basado en particionado k-means.

En este apartado, se va a facilitar imágenes del código utilizado.

```
2
3   import pandas as pd
4   import numpy as np
5   from sklearn.cluster import KMeans
6   from operator import itemgetter
7
8
9   def lectura_datos(fichero):
10      datos = pd.read_csv(fichero)
11      atrib = datos.values[1:,0:-1]
12      clases = datos.values[1:,-1]
13
14      return atrib, clases
15
16  def clustering(atributos, numCentroides):
17      centroides = atributos[np.random.choice(atributos.shape[0], numCentroides, replace = False)]
18      kmedias = KMeans(n_clusters = numCentroides, init = centroides, max_iter = 500, n_init = 1 , random_state=0)
19      dist = kmedias.fit_transform(atributos)
20      centros = kmedias.cluster_centers_
21
22      return kmedias, dist, centros
23
24  def calcular_SSE(distancias):
25      sum = 0
26      for i in range(distancias.shape[0]):
27          sum = sum + pow(np.min(distancias[i]),2)
28      return sum
29
```

Ilustración 1: Código k-means 1

```
if __name__ == '__main__':
    for fichero in ["diabetes","iris","ionosphere","segment-challenge","glass"]:
        nCentroides = 5
        semilla = 15
        np.random.seed(semilla)
        atrib, clases = lectura_datos(fichero+".csv")
        kmedias, distancias, centros = clustering(atrib, nCentroides)
        SSE = calcular_SSE(distancias)

        print ("\n"+fichero+"\n")

        print "Centroides: "
        print centros
        print "\nSSE: %f\n" %(SSE)
```

Ilustración 2: Código k-means 2

Ejercicio 4

Implemente la medida de evaluación de la calidad de un método de agrupación basada en SSE o en la correlación entre la matriz de incidenia y la de proximidad.

Detallado en el ejercicio anterior.

Ejercicio 5

Para cada uno de los problemas seleccionados realice las siguientes tareas:

5.1.- Ejecute el algoritmo de particionado y evalúe su rendimiento.

Base de Datos	SSE
diabetes	1966475,571483
iris	49,750247
ionosphere	1989,497587
Segmente-challenge	11950647,026016
glass	591,666841