Minería de Datos – 2019/2020

**Ejercicio entregable #4**

Luna Jiménez Fernández

Índice

[1. Ejercicio 1 3](#_Toc26312859)

[2. Ejercicio 2 8](#_Toc26312860)

[3. Ejercicio 3 13](#_Toc26312861)

# 1. Ejercicio 1

**Apartado A:**

Los valores de la variable X2 ordenados de menor a mayor son los siguientes:

**1, 1, 2, 2, 2.5, 3.5, 4, 4, 6, 7, 8, 10.**

Si se va a dividir en dos intervalos utilizando igual frecuencia, al tener doce elementos en el conjunto de entrenamiento lo que se hará es incluir 6 elementos en cada intervalo. Por tanto, los intervalos serán (ajustando el primer y último intervalo a menos infinito e infinito respectivamente):

**Intervalo 1:**

**Intervalo 2:**

Como se puede observar, hemos tomado para el principio y final de los intervalos 3.75, que es la media entre el 3.5 del último elemento que debería entrar en el intervalo 1 y el 4 del primer elemento que debería entrar en el intervalo 2.

**Apartado B:**

Para esta discretización, nos interesa observar los valores de X1 junto a sus valores de clase correspondiente:

|  |  |
| --- | --- |
| **X1** | **Clase** |
| 1 | azul |
| 1 | azul |
| 1 | rojo |
| 3 | azul |
| 3 | rojo |
| 3.5 | azul |
| 5 | rojo |
| 5 | azul |
| 5 | azul |
| 7 | rojo |
| 7 | azul |
| 9 | rojo |

En esta discretización, buscaremos el umbral que nos de la mayor **ganancia de información**.

**Primera partición:**

La entropía actual de la clase es:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Umbral** | **1** | **3** | **3.5** | **5** | **7** |
| **H(Subintervalo 1)** | 0.918 | 0.971 | 0.918 | 0.918 | 0.946 |
| **H(Subintervalo 2)** | 0.991 | 0.985 | 1 | 0.918 | 0 |
| **H(C|X1)** | 0.973 | 0.979 | 0.959 | 0.918 | 0,867 |
| **Ganancia de información** | 0.007 | 0.001 | 0.021 | 0.062 | **0,113** |

Se desarrollará un umbral para que se entienda el proceso. El resto de los umbrales no serán desarrollados para ahorrar espacio:

* X1 <= 1
  + Subintervalo 1: {azul, azul, rojo}. **Entropía:**
  + Subintervalo 2: {azul, rojo, azul, rojo, azul, azul, rojo, azul, rojo}. **Entropía**:
    - 1
  + **Entropía conocida la variable:**
  + **Ganancia de información:**

Como se observa en la tabla, el umbral que más **ganancia de información** nos ha dado es 7 (X1<=7). Por tanto, los intervalos que se tomarían serían los siguientes (tomando de nuevo un valor ligeramente superior al umbral para los intervalos):

**Intervalo 1:**

**Intervalo 2:**

**Segunda partición:**

Para añadir un tercer intervalo, tenemos que particionar alguno de los intervalos anteriores. El intervalo 2 no se puede particionar de nuevo (solo entra un caso en ese intervalo), por lo que particionaremos el intervalo 1. Los valores son los siguientes:

|  |  |
| --- | --- |
| **X1** | **Clase** |
| 1 | azul |
| 1 | azul |
| 1 | rojo |
| 3 | azul |
| 3 | rojo |
| 3.5 | azul |
| 5 | rojo |
| 5 | azul |
| 5 | azul |
| 7 | rojo |
| 7 | azul |

La entropía actual de la clase es la siguiente:

Las ganancias de información para cada posible umbral son las siguientes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Umbral** | **1** | **3** | **3.5** | **5** |
| **H(Subintervalo 1)** | 0.918 | 0.971 | 0.918 | 0.918 |
| **H(Subintervalo 2)** | 0.954 | 0.918 | 0.971 | 1 |
| **H(C|X1)** | 0.945 | 0.942 | 0.942 | 0.933 |
| **Ganancia de información** | 0.005 | 0.008 | 0.008 | **0.017** |

Como se observa en la tabla, el umbral que más **ganancia de información** nos ha dado es 5 (X1<=5). Por tanto, los intervalos que se tomarían serían los siguiente:

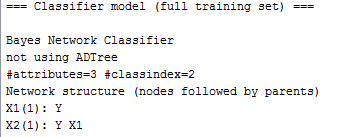
**Intervalo 1:**

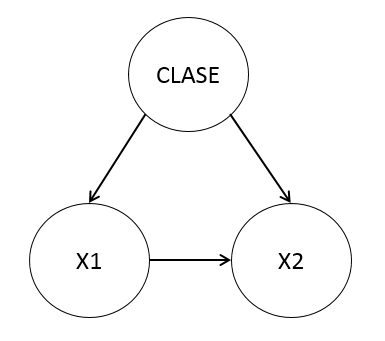
**Intervalo 2:** (5.5, 7.5]

**Intervalo 3:**

**Apartado C:**

Para obtener la **estructura** del TAN (Tree Augmented Naive Bayes), debemos hacer un árbol de peso máximo entre todas las variables predictoras. En este caso, al haber únicamente dos variables (X1 y X2), solo existirá un posible arco en el árbol (que se tomará).

Lo único que faltaría ahora sería decidir cuál es la variable que será la raíz del árbol (para orientar los arcos). La cabeza podría decidirse arbitrariamente, pero en este caso se ha utilizado Weka. Los resultados son los siguientes:

Como ahí se indica, en el árbol formado por las variables X1 y X2, X1 será el padre de X2. Por tanto, la estructura aprendida para el TAN será:

Lo único que queda ahora es aprender los parámetros de TAN utilizando **suavizado de Laplace:**

|  |  |  |
| --- | --- | --- |
| **P(Clase)** | **Clase = azul** | **Clase = rojo** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **P(X1|Clase)** | **Clase = azul** | **Clase = rojo** |
| **X1 =** |  |  |
| **X1 = (5.5, 7.5]** |  |  |
| **X1 =** |  |  |

Para la siguiente tabla, las correspondencias son:

* X1=1 → X1 =
* X1=2 → X1 = (5.5, 7.5]
* X1=3 → X1 =

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **P(X2|X1,Clase)** | **Clase = azul** | | | **Clase = rojo** | | |
| **X1=1** | **X1=2** | **X1=3** | **X1=1** | **X1=2** | **X1=3** |
| **X2 =** |  |  |  |  |  |  |
| **X2 =** |  |  |  |  |  |  |

**Apartado D:**

* **(2, 9, rojo)**

Discretizando, tendríamos que y **.**

Al ser mayor la probabilidad cuando la clase es azul, este caso se clasificará como **azul**. Esto supone un **error**.

* **(4, 5, rojo)**

Discretizando, tendríamos que y **.**

Al ser mayor la probabilidad cuando la clase es azul, este caso se clasificará como **azul**. Esto supone un **error**.

* **(8, 6, azul)**

Discretizando, tendríamos que y **.**

Al ser mayor la probabilidad cuando la clase es rojo, este caso se clasificará como **rojo**. Esto supone un **error**.

La tasa de acierto de este clasificador ha sido de **0%** respecto al conjunto de prueba.

**Apartado E:**

Para construir un clasificador TAN, son necesarias dos partes: especificar la **estructura** y aprender los **parámetros**.

Para especificar la **estructura**,es necesario conocer la **información mutua** **condicionada a la clase** de todos los pares de variables. En este caso, tendríamos dos variables predictoras (una discretizada y una continua), pero en principio es posible calcular esta información, aunque una de las variables sea continua.

Por otra parte, para aprender los **parámetros** se utiliza principalmente la probabilidad condicional (probabilidad de la variable condicionada a la de sus padres). De nuevo, esto sigue siendo posible independientemente de si la variable es discreta (la discretizada) o numérica. Para tratar la numérica, simplemente se trataría como una **distribución normal**.

Por lo tanto, en principio **sería posible construir el clasificador TAN** en este caso, aunque una de las variables estuviese sin discretizar.

# 2. Ejercicio 2

Para mayor claridad, se presenta una tabla con la **incertidumbre simétrica** (SU) de las cinco variables predictoras y la clase:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| **Y** | 0.134 | 0.350 | 0.063 | 0.386 | 0.005 |
| **A** |  | 0.001 | 0.009 | 0.001 | 0.003 |
| **B** |  |  | 0.012 | 0.011 | 0.002 |
| **C** |  |  |  | 0.012 | 0.000 |
| **D** |  |  |  |  | 0.011 |

**Apartado A:**

En este apartado, se identificará el subconjunto de variables seleccionadas utilizando un método **forward** (empezamos sin ninguna variable y vamos añadiendo variables de una en una, buscando añadir la que maximiza el evaluador) utilizando la técnica de **CFS**.

La fórmula a utilizar es la siguiente:

El conjunto inicial de variables elegidas es el conjunto vacío, por lo que el evaluador devolverá:

**Primera variable:**

Hay que elegir cuál de las cinco variables predictoras se elegirá primero. En este caso, tendremos una variable (n=1).

La variable que mejora más el evaluador si es añadida es **D**. Por tanto, esta será la primera variable añadida. Las variables seleccionadas actualmente son **{D}**, dando una evaluación de **0.386**.

**Segunda variable:**

Hay que elegir cuál de las siguientes cuatro variables predictoras se elegirá. Tendremos dos variables (n=2)

La variable que más mejora el evaluador al ser añadida es **B**. Además, la nueva evaluación (0.518) mejora a la anterior (0.386). Por tanto, esta variable será añadida. Las variables seleccionadas actualmente son **{D,B}** con una evaluación de **0.518**.

**Tercera variable:**

Hay que elegir cuál de las siguientes tres variables predictoras se elegirá. Tendremos tres variables (n=3)

La variable que más mejora el evaluador al ser añadida es **A**. A pesar de esto, la nueva evaluación (0.500) **no** mejora a la anterior (0.518). Por tanto, esta variable no será añadida.

Finalmente, las variables seleccionadas son **{D,B}** con una evaluación de **0.518**.

**Apartado B:**

En este apartado, se identificará el subconjunto de variables seleccionadas utilizando un método **backwards** (empezamos con todas las variables y las vamos eliminando de una en una, buscando maximizar el evaluador) utilizando la técnica de **MIFS**.

La fórmula a utilizar es la siguiente:

Para el valor de **beta** utilizaremos su valor por defecto (0.5)

El conjunto inicial de variables elegidas es el conjunto de todas las variables predictoras, por lo que el evaluador devolverá:

**Primera variable:**

Hay que elegir cuál de las cinco variables predictoras se eliminará primero:

La variable que más mejora el clasificador al ser eliminada es **E**. Además, el evaluador obtenido eliminando la variable (0.91) es mejor al obtenido previamente (0.907)

Por tanto, se eliminará la variable **E**. Las variables restantes son **{A,B,C,D}** con una evaluación de **0.91**.

**Segunda variable:**

Hay que elegir cuál de las cuatro variables predictoras restantes se eliminará:

La variable que mejor evaluación presenta al ser eliminada es la **C**. En cambio, esta nueva evaluación (0.864) **no** mejora respecto a la anterior (0.91). Por lo tanto, **no** se eliminará ninguna variable.

Finalmente, las variables seleccionadas son **{A,B,C,D}** con una evaluación de **0.91**.

**Apartado C:**

En este apartado, se utilizará el método **Ranker**. Este método consiste en ordenar las variables según una métrica univariada, y después elegir una cantidad de ellas (en orden, a partir de la primera variable)

En este caso, utilizaremos la **incertidumbre simétrica** con la clase como métrica (ya que es la que se nos ha proporcionado). Las cinco variables ordenadas son:

1. D (0.386)
2. B (0.350)
3. A (0.134)
4. C (0.063)
5. E (0.005)

Si elegimos el mismo número de variables que las que hemos obtenido utilizando CFS y MIFS, los conjuntos que se elegirán son:

**CFS (2 variables):** {D,B}

**MIFS (4 variables):** {D,B,A,C}

Como se puede observar, estos conjuntos son equivalentes a los obtenidos en los dos apartados anteriores.

**Apartado D:**

De los apartados anteriores, hemos obtenido dos subconjuntos:

* **{D,B}**, obtenido con CFS utilizando un método **forward**.
* **{A,B,C,D}**, obtenido con MIFS utilizando un método **backward**.

Lo primero y más evidente es que ambos subconjuntos **son distintos**. Esto se debe seguramente a que ambos conjuntos se corresponden con **óptimos locales distintos**. Al haber empezado ambos métodos desde puntos distintos, es normal que los óptimos que han alcanzado sean distintos.

Aun así, ambos subconjuntos contienen las variables **B** y **D**. Por tanto, podemos pensar que estas variables son las que mayor capacidad discriminatoria tiene. También se ve que ninguno de los dos subconjuntos contiene la variable **E**, por lo que probablemente no era útil a la hora de clasificar Y.

El subconjunto obtenido por backward contiene además las variables **A** y **C**. Es posible que ambas variables discriminen bien la clase cuando trabajan en conjunto y por eso se hayan conservado en backward (mientras que como forward añade las variables de una en una no pudiese observar esa mejora cuando están ambas variables presentes a la vez, ya que las prueba por separado).

Finalmente, se puede ver que el subconjunto obtenido por forward es **más pequeño** que el obtenido por backward. De nuevo, esto tiene sentido: en el método forward, empezamos con un conjunto de variables vacío y las vamos añadiendo una a una mientras mejore; mientras que en el método backward empezamos con todas las variables en el conjunto y las vamos eliminando una a una mientras mejore.

Por tanto, suponiendo que ambos encuentren un óptimo local pronto, el método forward habrá añadido pocas variables (y por tanto el conjunto resultante será pequeño) mientras que el método backward habrá eliminado pocas variables (y por tanto el conjunto resultante será grande)

# 3. Ejercicio 3

**Apartado A:**

Para obtener las tres muestras, utilizaremos **muestreo con reemplazo** (las instancias no se eliminan del conjunto de datos cuando se seleccionan, por lo que una misma instancia puede aparecer varias veces).

El muestreo debe hacerse de forma aleatoria. Por tanto, lo que haremos será ir añadiendo a la muestra la instancia que se corresponda con el siguiente número del generador aleatorio:

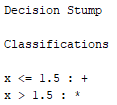
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Muestra 1** | | | | **Muestra 2** | | | | **Muestra 3** | | | |
| **ID** | **X** | **Y** | **Clase** | **ID** | **X** | **Y** | **Clase** | **ID** | **X** | **Y** | **Clase** |
| **1** | 1 | 1 | **+** | **3** | 2 | 1 | **\*** | **1** | 1 | 1 | **+** |
| **2** | 1 | 3 | **+** | **5** | 3 | 1 | **\*** | **2** | 1 | 3 | **+** |
| **3** | 2 | 1 | **\*** | **8** | 4 | 3 | **+** | **3** | 2 | 1 | **\*** |
| **4** | 2 | 3 | **\*** | **6** | 3 | 3 | **\*** | **4** | 2 | 3 | **\*** |
| **5** | 3 | 1 | **\*** | **4** | 2 | 3 | **\*** | **7** | 4 | 1 | **+** |
| **6** | 3 | 3 | **\*** | **2** | 1 | 3 | **+** | **8** | 4 | 3 | **+** |
| **1** | 1 | 1 | **+** | **3** | 2 | 1 | **\*** | **1** | 1 | 1 | **+** |
| **2** | 1 | 3 | **+** | **5** | 3 | 1 | **\*** | **2** | 1 | 3 | **+** |

**Apartado B:**

Para **bagging**, se realizará una iteración para cada muestra, aprendiendo un clasificador distinto en cada una de ellas. En este caso, aprenderemos **decision stumps** en todas las iteraciones.

Para simplificar los cálculos, utilizaremos Weka para obtener los decision stumps óptimos de cada muestra, aunque también podrían obtenerse estos *a ojo*.

**Iteración 1 (Muestra 1):**

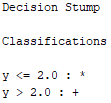
El **decision stump** devuelto por Weka para la muestra es el siguiente:

Por tanto, el modelo aprendido **(Modelo 1)** es:

**If X <= 1.5 then +**

**Else \***

**Iteración 2 (Muestra 2):**

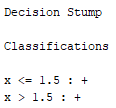
El **decision stump** devuelto por Weka para la muestra es el siguiente:

Por tanto, el modelo aprendido (**Modelo 2)** es:

**If Y <= 2 then \***

**Else +**

**Iteración 3 (Muestra 3):**

El **decision stump** devuelto por Weka para la muestra es el siguiente:

Por tanto, el modelo aprendido **(Modelo 3)** es:

**If X <= 1.5 then +**

**Else \***

**Resultados con el conjunto de datos:**

Ahora comprobaremos el **accuracy** de cada modelo con el conjunto de entrenamiento entero, y posteriormente comprobaremos el accuracy del ensemble entero con el conjunto de entrenamiento.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modelo 1** | | | | | |
| **ID** | **X** | **Y** | **Clase esperada** | **Clase obtenida** | **Acierto/ Fallo** |
| 1 | 1 | 1 | **+** | **+** | Acierto |
| 2 | 1 | 3 | **+** | **+** | Acierto |
| 3 | 2 | 1 | **\*** | **\*** | Acierto |
| 4 | 2 | 3 | **\*** | **\*** | Acierto |
| 5 | 3 | 1 | **\*** | **\*** | Acierto |
| 6 | 3 | 3 | **\*** | **\*** | Acierto |
| 7 | 4 | 1 | **+** | **\*** | Fallo |
| 8 | 4 | 3 | **+** | **\*** | Fallo |

El modelo 1 tiene un **accuracy** de

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modelo 2** | | | | | |
| **ID** | **X** | **Y** | **Clase esperada** | **Clase obtenida** | **Acierto/ Fallo** |
| 1 | 1 | 1 | **+** | **\*** | Fallo |
| 2 | 1 | 3 | **+** | **+** | Acierto |
| 3 | 2 | 1 | **\*** | **\*** | Acierto |
| 4 | 2 | 3 | **\*** | **+** | Fallo |
| 5 | 3 | 1 | **\*** | **\*** | Acierto |
| 6 | 3 | 3 | **\*** | **+** | Fallo |
| 7 | 4 | 1 | **+** | **\*** | Fallo |
| 8 | 4 | 3 | **+** | **+** | Acierto |

El modelo 2 tiene un **accuracy** de

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modelo 3** | | | | | |
| **ID** | **X** | **Y** | **Clase esperada** | **Clase obtenida** | **Acierto/ Fallo** |
| 1 | 1 | 1 | **+** | **+** | Acierto |
| 2 | 1 | 3 | **+** | **+** | Acierto |
| 3 | 2 | 1 | **\*** | **\*** | Acierto |
| 4 | 2 | 3 | **\*** | **\*** | Acierto |
| 5 | 3 | 1 | **\*** | **\*** | Acierto |
| 6 | 3 | 3 | **\*** | **\*** | Acierto |
| 7 | 4 | 1 | **+** | **\*** | Fallo |
| 8 | 4 | 3 | **+** | **\*** | Fallo |

El modelo 3 tiene un **accuracy** de

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ensemble** | | | | | | | | |
| **ID** | **X** | **Y** | **Clase esperada** | **Clase Modelo 1** | **Clase Modelo 2** | **Clase Modelo 3** | **Clase Mayoría** | **Acierto/ Fallo** |
| 1 | 1 | 1 | **+** | **+** | **\*** | **+** | **+** | Acierto |
| 2 | 1 | 3 | **+** | **+** | **+** | **+** | **+** | Acierto |
| 3 | 2 | 1 | **\*** | **\*** | **\*** | **\*** | **\*** | Acierto |
| 4 | 2 | 3 | **\*** | **\*** | **+** | **\*** | **\*** | Acierto |
| 5 | 3 | 1 | **\*** | **\*** | **\*** | **\*** | **\*** | Acierto |
| 6 | 3 | 3 | **\*** | **\*** | **+** | **\*** | **\*** | Acierto |
| 7 | 4 | 1 | **+** | **\*** | **\*** | **\*** | **\*** | Fallo |
| 8 | 4 | 3 | **+** | **\*** | **+** | **\*** | **\*** | Fallo |

Finalmente, el **accuracy** del ensemble para todo el conjunto de entrenamiento es de **.**

Este accuracy no mejora realmente al del modelo 1 o 3 para todo el conjunto de entrenamiento, pero esto se debe a que el ensemble elige la clase de la entrada usando voto por mayoría. Como los modelos 1 y 3 son iguales y únicamente hay 3 modelos en el ensemble, la mayoría (y por tanto el ensemble) elegirá lo mismo que esos modelos.

**Apartado C:**

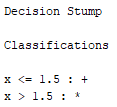
En este caso realizaremos **boosting**, concretamente AdaBoost. Para este método utilizaremos el conjunto de entrenamiento directamente sin muestreo, y en cada iteración aprenderemos un modelo basado en el modelo que aprendimos en la iteración anterior.

De nuevo, los modelos que aprenderemos serán **decision stumps**. Otra vez utilizaremos Weka, aunque sería posible aprender estos arboles *a ojo* teniendo en cuenta los pesos de los casos.

Lo primero es describir primero los pesos de los casos del conjunto de entrenamiento:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **X** | **Y** | **Clase** | **Peso** |
| **1** | 1 | 1 | **+** | 0.125 |
| **2** | 1 | 3 | **+** | 0.125 |
| **3** | 2 | 1 | **\*** | 0.125 |
| **4** | 2 | 3 | **\*** | 0.125 |
| **5** | 3 | 1 | **\*** | 0.125 |
| **6** | 3 | 3 | **\*** | 0.125 |
| **7** | 4 | 1 | **+** | 0.125 |
| **8** | 4 | 3 | **+** | 0.125 |

**Iteración 1:**

El modelo aprendido por Weka para esta iteración es el siguiente:

**If X <= 1 then +**

**Else \***

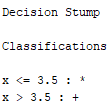
Ahora podemos calcular la clase obtenida con el nuevo modelo para cada uno de los casos, y (posteriormente) calcular los pesos ajustados:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Iteración 1** | | | | | | | | |
| **ID** | **X** | **Y** | **Clase esperada** | **Peso original** | **Clase obtenida** | **Acierto/ Fallo** | **Peso ajustado** | **Peso normalizado** |
| 1 | 1 | 1 | **+** | 0.125 | **+** | Acierto |  | 0.083 |
| 2 | 1 | 3 | **+** | 0.125 | **+** | Acierto |  | 0.083 |
| 3 | 2 | 1 | **\*** | 0.125 | **\*** | Acierto |  | 0.083 |
| 4 | 2 | 3 | **\*** | 0.125 | **\*** | Acierto |  | 0.083 |
| 5 | 3 | 1 | **\*** | 0.125 | **\*** | Acierto |  | 0.083 |
| 6 | 3 | 3 | **\*** | 0.125 | **\*** | Acierto |  | 0.083 |
| 7 | 4 | 1 | **+** | 0.125 | **\*** | Fallo |  | 0.25 |
| 8 | 4 | 3 | **+** | 0.125 | **\*** | Fallo |  | 0.25 |

De estos resultados, podemos calcular las siguientes métricas del modelo:

Tras calcular los pesos ajustados, es posible calcular la **constante de normalización** para normalizar los pesos (siendo esta constante la suma de todos los pesos sin ajustar):

**Iteración 2:**

El modelo aprendido por Weka para esta iteración (teniendo en cuenta los nuevos pesos de las instancias calculadas previamente) es el siguiente:

**If X <= 3 then \***

**Else +**

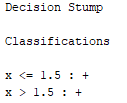
Ahora podemos calcular la clase obtenida con el nuevo modelo para cada uno de los casos, y (posteriormente) calcular los pesos ajustados:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Iteración 2** | | | | | | | | |
| **ID** | **X** | **Y** | **Clase esperada** | **Peso original** | **Clase obtenida** | **Acierto/ Fallo** | **Peso ajustado** | **Peso normalizado** |
| 1 | 1 | 1 | **+** | 0.083 | **\*** | Fallo |  | 0.25 |
| 2 | 1 | 3 | **+** | 0.083 | **\*** | Fallo |  | 0.25 |
| 3 | 2 | 1 | **\*** | 0.083 | **\*** | Acierto |  | 0.05 |
| 4 | 2 | 3 | **\*** | 0.083 | **\*** | Acierto |  | 0.05 |
| 5 | 3 | 1 | **\*** | 0.083 | **\*** | Acierto |  | 0.05 |
| 6 | 3 | 3 | **\*** | 0.083 | **\*** | Acierto |  | 0.05 |
| 7 | 4 | 1 | **+** | 0.25 | **+** | Acierto |  | 0.15 |
| 8 | 4 | 3 | **+** | 0.25 | **+** | Acierto |  | 0.15 |

De estos resultados, podemos calcular las siguientes métricas del modelo (asumimos que la suma de todos los pesos normalizados es 1):

Tras calcular los pesos ajustados, es posible calcular la **constante de normalización** para normalizar los pesos:

**Iteración 3:**

El modelo aprendido por Weka para esta iteración (teniendo en cuenta los nuevos pesos de las instancias calculadas previamente) es el siguiente:

**If X <= 1 then +**

**Else +**

Esto equivaldría al siguiente ZeroR:

**\_ +**

Ahora podemos calcular la clase obtenida con el nuevo modelo. Al ser la última iteración, no será necesario calcular nuevos pesos.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **X** | **Y** | **Clase esperada** | **Peso original** | **Clase obtenida** | **Acierto/ Fallo** |
| 1 | 1 | 1 | **+** | 0.25 | **+** | Acierto |
| 2 | 1 | 3 | **+** | 0.25 | **+** | Acierto |
| 3 | 2 | 1 | **\*** | 0.05 | **+** | Fallo |
| 4 | 2 | 3 | **\*** | 0.05 | **+** | Fallo |
| 5 | 3 | 1 | **\*** | 0.05 | **+** | Fallo |
| 6 | 3 | 3 | **\*** | 0.05 | **+** | Fallo |
| 7 | 4 | 1 | **+** | 0.15 | **+** | Acierto |
| 8 | 4 | 3 | **+** | 0.15 | **+** | Acierto |

De estos resultados, podemos calcular las siguientes métricas del modelo:

**Accuracy del ensemble**

Finalmente, se calculará el accuracy del ensemble con el conjunto de entrenamiento. Para esto, se utilizará un **voto normalizado por el peso** de cada modelo del ensemble. La clase que tenga un valor superior será la clase elegida por el ensemble. Toda la información necesaria está recogida en la siguiente tabla:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Accuracy** | | | | | | | | | | |
| **ID** | **X** | **Y** | **Clase esperada** | **Clase Modelo 1** | **Clase Modelo 2** | **Clase Modelo 3** | **Peso Clase +** | **Peso Clase \*** | **Clase Ensemble** | **Acierto/Fallo** |
| 1 | 1 | 1 | **+** | **+** | **\*** | **+** | **1.242** | 0.807 | **+** | Acierto |
| 2 | 1 | 3 | **+** | **+** | **\*** | **+** | **1.242** | 0.807 | **+** | Acierto |
| 3 | 2 | 1 | **\*** | **\*** | **\*** | **+** | 0.693 | **1.356** | **\*** | Acierto |
| 4 | 2 | 3 | **\*** | **\*** | **\*** | **+** | 0.693 | **1.356** | **\*** | Acierto |
| 5 | 3 | 1 | **\*** | **\*** | **\*** | **+** | 0.693 | **1.356** | **\*** | Acierto |
| 6 | 3 | 3 | **\*** | **\*** | **\*** | **+** | 0.693 | **1.356** | **\*** | Acierto |
| 7 | 4 | 1 | **+** | **\*** | **+** | **+** | **1.5** | 0.549 | **+** | Acierto |
| 8 | 4 | 3 | **+** | **\*** | **+** | **+** | **1.5** | 0.549 | **+** | Acierto |

La importancia de cada clasificador es la siguiente:

**Modelo 1:** 0.549

**Modelo 2:** 0.807

**Modelo 3:** 0.693

Además, el cálculo del peso para cada clase en cada instancia en el ensemble es el siguiente:

|  |  |  |
| --- | --- | --- |
| **ID** | **Peso Clase +** | **Peso Clase \*** |
| **1** | 0.549+0.693 = 1.242 | 0.807 |
| **2** | 0.549+0.693 = 1.242 | 0.807 |
| **3** | 0.693 | 0.549+0.807 = 1.356 |
| **4** | 0.693 | 0.549+0.807 = 1.356 |
| **5** | 0.693 | 0.549+0.807 = 1.356 |
| **6** | 0.693 | 0.549+0.807 = 1.356 |
| **1** | 0.807+0.693 = 1.5 | 0.549 |
| **2** | 0.807+0.693 = 1.5 | 0.549 |

El **accuracy** de este ensemble es del **100%**, ya que acierta (clasifica correctamente) todos los casos del conjunto de entrenamiento.