

Contenido

Analizando las variables	2
Pre-procesando los datos.....	3
Limpieza de los datos	3
Aplicando transformaciones	6
Selección de variables	7
Balanceo de datos	8
Aplicando métodos supervisados	9
Métodos de regresión	9
Probando el modelo	11
Métodos basados en ejemplos	13
Optimizando los parámetros.....	13
Probando el método con nuevos datos	16
Máquina de soporte vectorial	17
Optimizando los parámetros.....	17
Probando el método con nuevos datos	19
Arboles de decisión	19
Optimizando los parámetros.....	20
Probando el método con nuevos datos	25
Bonus.....	26
Método Bayesiano	26
Probando el método con datos nuevos	29
Redes neuronales.....	29
Observaciones	35
Probando el método con datos nuevos	39
Conclusiones, Eligiendo a los ganadores.....	39

Autor: José Romualdo Villalobos Pérez

ID: 000294087

Lo primero que se hizo fue agregar el encabezado de WEKA al archivo drug.csv y se guardó como drug.arff

Analizando las variables

La variable edad tiene una distribución normal, se observó que hay un individuo con una edad de 145 lo cual se puede considerar bastante improbable ya que es un dato atípico, por lo tanto, se procederá a eliminar este dato.

La variable sexo esta balanceada ya que 52% son hombres y el otro 48% son mujeres.

VALOR VARIABLE SEX	PORCENTAJE PERSONAS CON EL MISMO VALOR
M	52%
F	48%

La variable BP (Blood Pressure) También se encuentra balanceada:

VALOR VARIABLE BP	PORCENTAJE PERSONAS CON EL MISMO VALOR
LOW	32%
NORMAL	29.5%
HIGH	38.5%

Respecto a la variable **cholesterol** se encontró un detalle, nadie tenía el colesterol en LOW por lo tanto se pensó una posible variable desbalanceada, así que se decidió consultar en portales médicos si es posible tener el colesterol bajo, se encontró lo siguiente:

“Los médicos siguen tratando de saber más sobre la conexión entre el colesterol bajo y los riesgos para la salud. No hay consenso respecto de cómo definir un nivel muy bajo de colesterol LDL, pero el nivel de LDL se consideraría muy bajo si está por debajo de 40 miligramos por decilitro de sangre.”

Debido a que es un tema que aun se encuentra en discusión se considera para este ejercicio que esta variable categoría se encuentra balanceada ya que solo puede tener dos valores NORMAL o HIGH.

VALOR VARIABLE CHOLESTEROL	PORCENTAJE PERSONAS CON EL MISMO VALOR
NORMAL	48.5%
HIGH	51.5%

Por lo tanto, se concluye que esta variable se encuentra balanceada.

En la variable continua **Na** se encontró que hay un dato atípico, una persona con un nivel de sodio de -0.85, se sospecha que este valor no es humanamente posible (al menos que se este en la condición incapacitante de la muerte) aunque no se encontró una fuente médica confiable que permitiera comprobar esta afirmación, por lo tanto, se procederá a eliminar a esta persona de los registros.

Al analizar la variable continua **K** se encontró que tiene una distribución uniforme y no se vio ningún valor anormal.

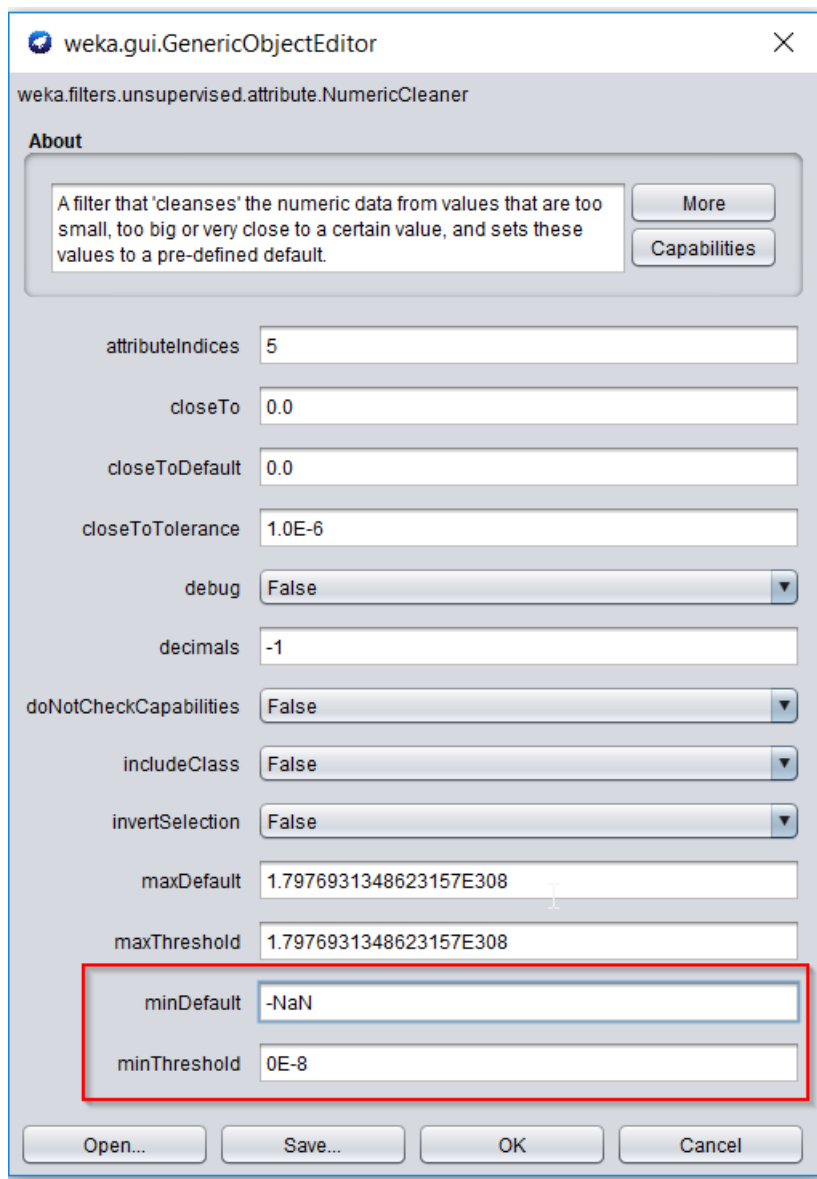
Pre-procesando los datos

Limpieza de los datos

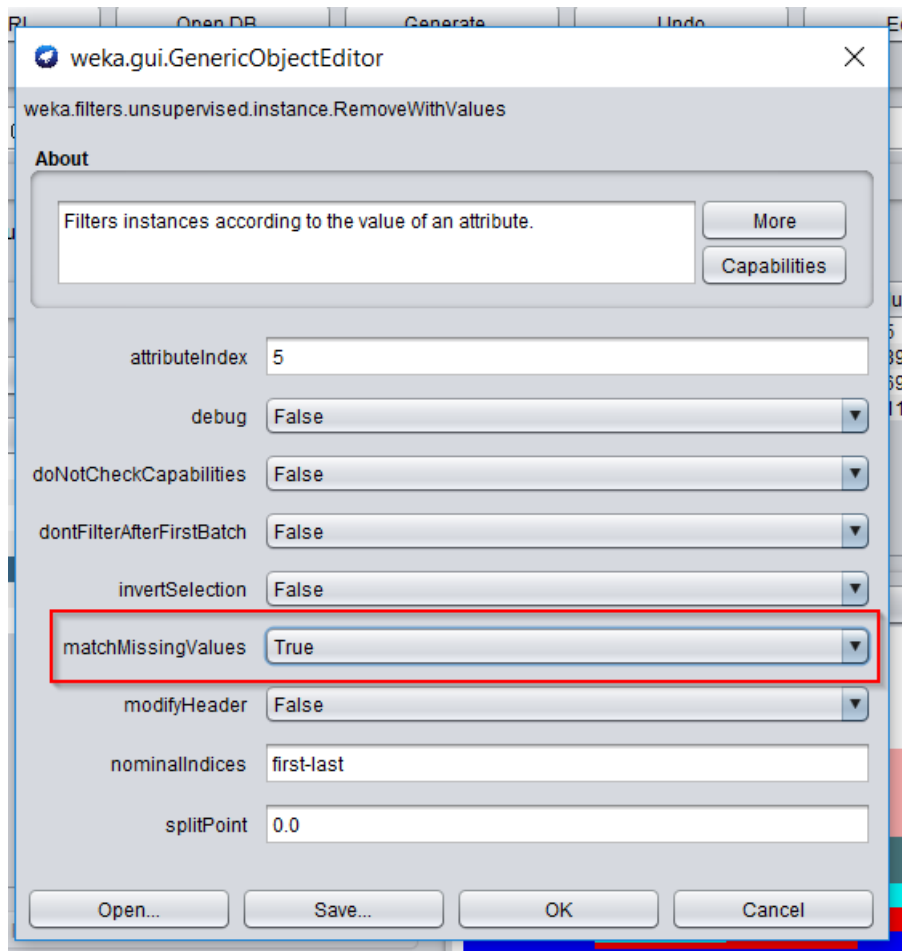
Debido a que hay poquitos pacientes (instancias) con valores atípicos se decidió eliminar dichas instancias en lugar de reemplazar el valor del atributo en cuestión.

Para lograrlo se hizo lo siguiente:

Para la variable **Na** se aplicó el filtro **NumericCleaner** y se marcaron como valores perdidos aquellos que estuvieran por debajo de 0 es decir se marcaron como NaN.



Luego se aplicó un filtro que elimina cualquier instancia que tenga un atributo con un valor perdido.



Luego de aplicar este filtro se pudo observar que el número de instancias bajo de 200 a 199, se aplicó un proceso similar para eliminar la instancia con la edad atípica, solo que en este caso se puso el maxThreshold del NumericCleaner en 110 (ya que es poco probable que algún humano pase esa edad)

Siguiendo con la limpieza de datos, no se encontró ni datos ausentes o nulos ni registros repetidos, razón por la cual se continuó al siguiente paso del preprocesamiento de los datos.

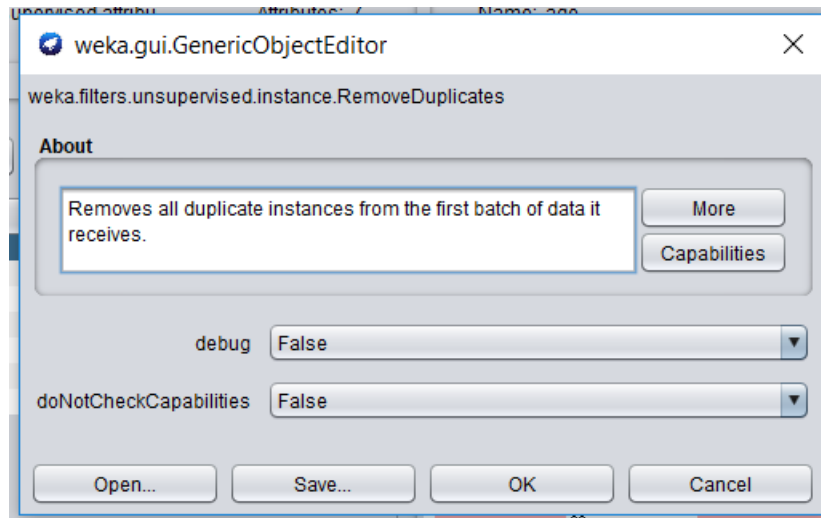
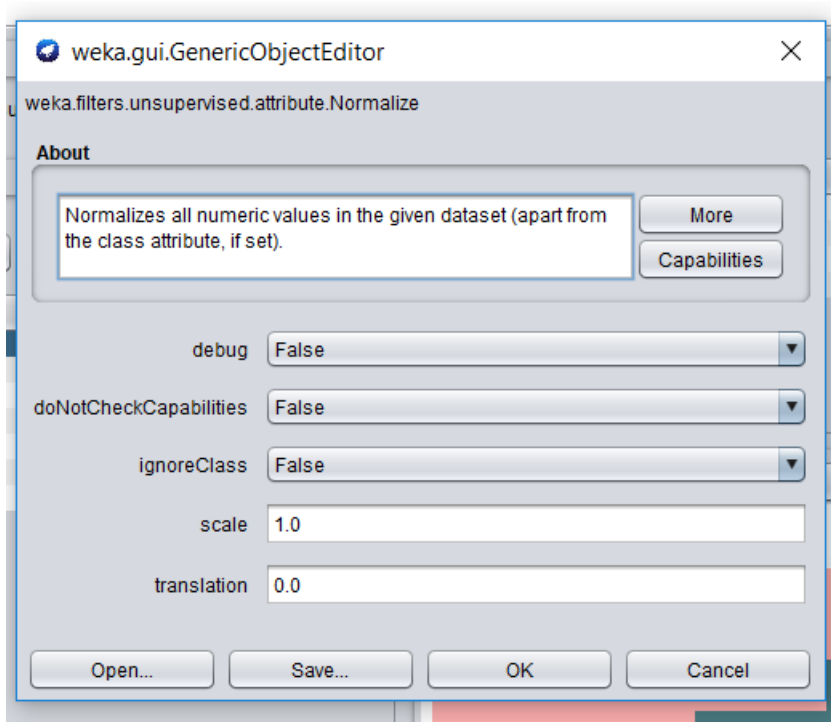


Ilustración 1 Después de aplicar un filtro que remueve los duplicados se observó que no se eliminó ninguna instancia, por tanto se concluye que no hay registros repetidos.

Volviendo a observar las variables continuas se observa todas tienen una distribución uniforme.

Aplicando transformaciones

Se procederá a continuación a normalizar todas las variables continuas para asegurar que los entrenamientos no sean muy sensibles a las diferencias de escala entre los valores de los atributos. Por ejemplo: Si observamos los valores de los atributos **Age** y **Na** veremos que el primero está en el rango [15, 74] y el segundo en el rango [0.5, 0.896]



No se discretizarán los atributos con valores continuos en este punto.

Selección de variables

Se procederá ahora a buscar los atributos con mayor importancia usando Principal Component Analysis este algoritmo puede crear atributos nuevos combinando los atributos originales.

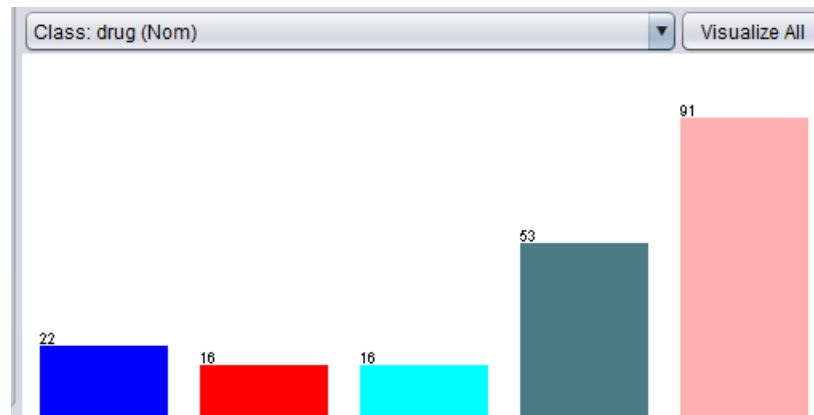
```
Ranked attributes:
0.7917331009423663 1 0.708BP=HIGH-0.518BP=NORMAL-0.23BP=LOW-0.229cholesterol=HIGH+0.222Na...
0.6015619001992742 2 0.748BP=LOW-0.548BP=NORMAL+0.202age-0.201BP=HIGH+0.175Na...
0.4572687656456079 3 0.6 age+0.549K-0.443sex=F+0.231BP=HIGH+0.195Na...
0.3181355879515144 4 -0.659Na-0.585sex=F-0.394cholesterol=HIGH-0.224age-0.098BP=NORMAL...
0.19377973263209636 5 -0.691cholesterol=HIGH+0.571K+0.324sex=F+0.192Na-0.15age...
0.09329243916633545 6 0.656age-0.497K-0.496cholesterol=HIGH+0.175BP=NORMAL-0.135BP=LOW...
0.00000000000000222 7 -0.632Na+0.563sex=F+0.291K+0.258age-0.244BP=NORMAL...

Selected attributes: 1,2,3,4,5,6,7 : 7
```

Se encontró un problema subjetivo al aplicar este método. Los atributos se vuelven mucho más difíciles de entender por un humano, lo cual hace difícil usar el sistema con nuevas instancias de ejemplo.

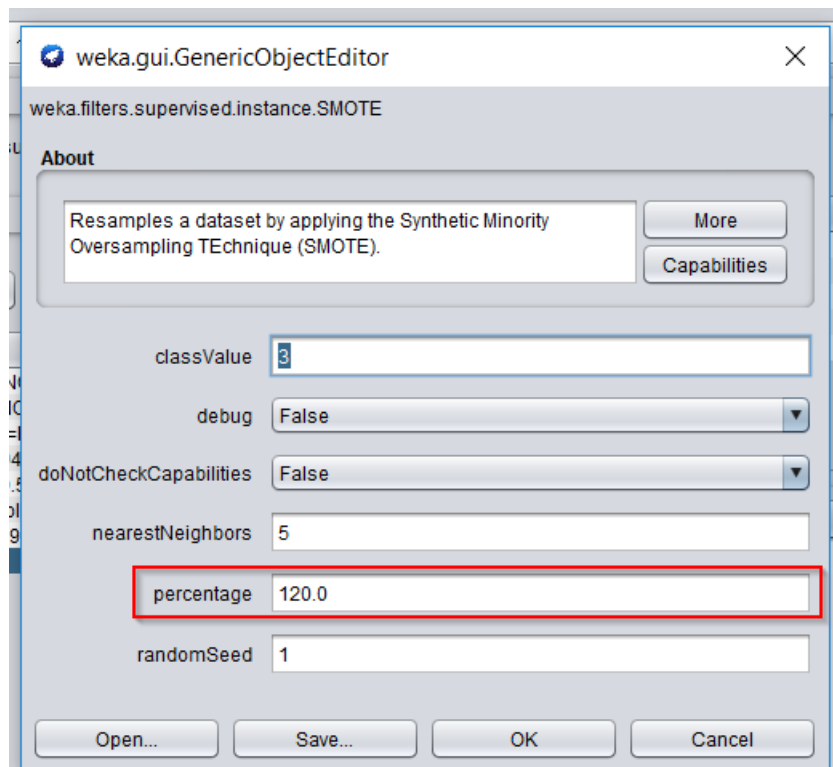
Además, debido a la relativamente baja cantidad de atributos se considero que no hace falta eliminar atributos.

Balanceo de datos

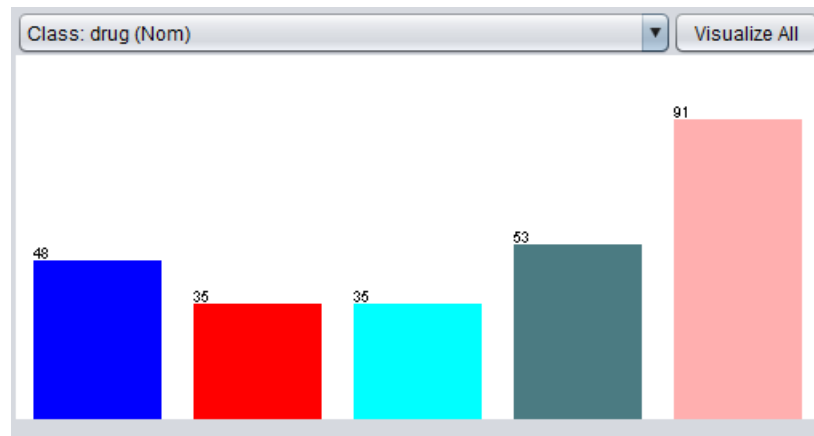


Al observar la **class** se observa que esta desbalanceada, lo cual nos expone a un alto riesgo de overfitting, razón por la cual se procedió a balancearlos “adicionando instancias” usando la técnica **SMOTE (Synthetic Minority Over-sampling Technique)**.

Se aplicó el **SMOTE** a los valores drugA, drugB y drugC de la **class**, para ello se usaron los siguientes parámetros de configuración:



Se incremento en un 120% la cantidad de instancias cuyas clases tienen valores drugA, drugB y drugC. El resultado obtenido fue el siguiente:

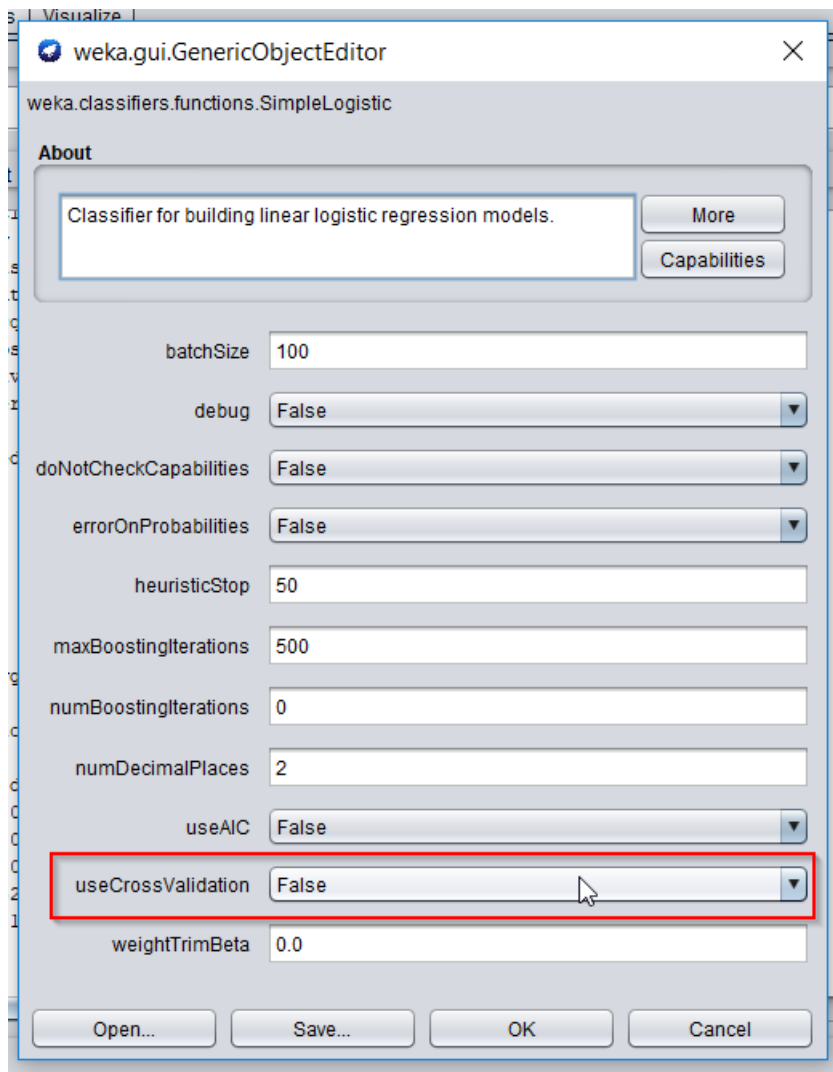


Cabe mencionar que este proceso incremento el numero de instancias de 198 a 262.

Aplicando métodos supervisados

Métodos de regresión

No se usará el método de regresión lineal porque el atributo a predecir es categórico en su lugar se usará el método de regresión logística el cual se usa cuando el atributo a predecir es categórico.



Se desactivo la opción que viene incorporada en el método SimpleLogistic de realizar validación cruzada porque para esta labor se usará la opción que viene incorporada por defecto en weka.

Debido a que este es un problema de regresión con múltiples clases nuestro modelo tendrá n regresiones donde n es el número de clases diferentes, a continuación, se muestra el modelo obtenido:

SimpleLogistic:

```
Class drugA :  
-5.01 +  
[age] * -12.54 +  
[sex=F] * -0.61 +  
[BP=HIGH] * 10.18 +  
[cholesterol=HIGH] * 3.23 +  
[Na] * -6.07 +  
[K] * 5.13
```

```
Class drugB :  
-25.94 +  
[age] * 17.78 +  
[sex=F] * 2.12 +  
[BP=HIGH] * 9.55 +  
[cholesterol=HIGH] * 3.65 +  
[K] * 4.02
```

```
Class drugC :  
-20.57 +  
[age] * -1.63 +  
[sex=F] * 1.61 +  
[BP=LOW] * 11.58 +  
[cholesterol=HIGH] * 9.39 +  
[Na] * -10.62 +  
[K] * 14.64
```

```
Class drugX :  
1.98 +  
[age] * 0.69 +  
[sex=F] * 1.17 +  
[BP=NORMAL] * 3.36 +  
[BP=HIGH] * -10.63 +  
[cholesterol=HIGH] * -2.9 +  
[Na] * -7.97 +  
[K] * 5.9
```

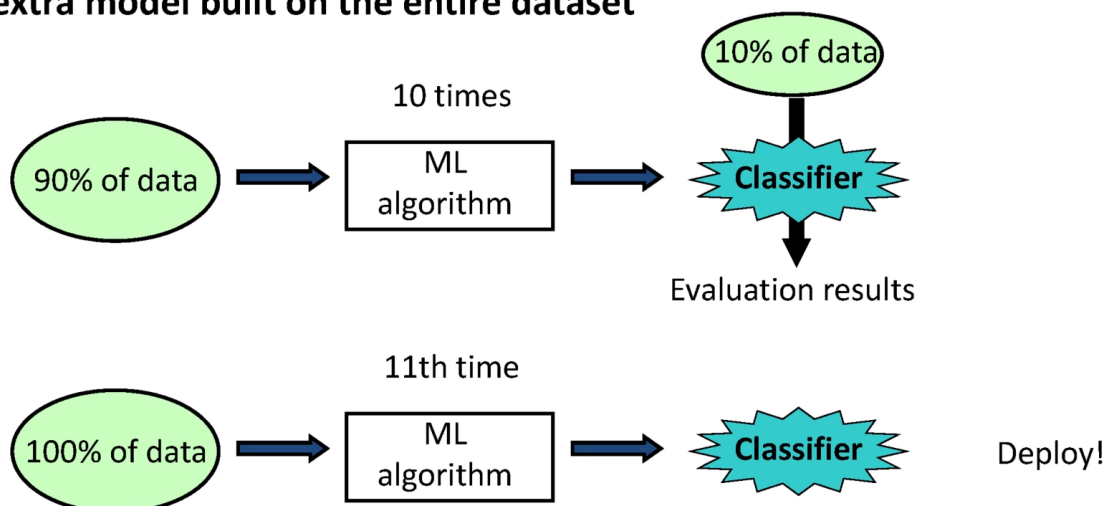
```
Class drugY :  
17.38 +  
[Na] * 25.63 +  
[K] * -60.61
```

Para una instancia de ejemplo determinada nosotros escogeremos la clase cuyo output sea el más grande (la de mayor probabilidad).

Probando el modelo

Debido a que el dataset es relativamente pequeño se decidió probar el modelo usando el método de cross validation

After cross-validation, Weka outputs an extra model built on the entire dataset



Se realizó un **stratified-cross-validation** con un **fold** de 10, nótese que por defecto Weka realiza una validación cruzada estratificada lo cual asegura que cuando se realizan los **folds** los valores de las clases tengan una proporción similar a los del **dataset** original.

A continuación, se muestra los resultados después de probar el modelo obtenido con 26 instancias que no se usaron para entrenar el modelo.

Instance | Actual | Predicted | Error | Probability

1	5:drugY	5:drugY	1
2	5:drugY	5:drugY	1
3	5:drugY	5:drugY	0.991
4	5:drugY	5:drugY	1
5	5:drugY	5:drugY	1
6	5:drugY	5:drugY	1
7	5:drugY	5:drugY	0.999
8	5:drugY	5:drugY	1
9	5:drugY	5:drugY	1
10	4:drugX	4:drugX	0.979
11	4:drugX	4:drugX	1
12	4:drugX	4:drugX	0.995
13	4:drugX	4:drugX	0.764
14	4:drugX	5:drugY +	0.568
15	3:drugC	3:drugC	0.964
16	3:drugC	3:drugC	0.996
17	3:drugC	3:drugC	0.977
18	1:drugA	1:drugA	0.956
19	1:drugA	1:drugA	0.926
20	1:drugA	1:drugA	0.993
21	1:drugA	1:drugA	0.995
22	1:drugA	1:drugA	0.998
23	2:drugB	2:drugB	0.992
24	2:drugB	2:drugB	0.994
25	2:drugB	2:drugB	0.829
26	2:drugB	2:drugB	0.984

La efectividad:

Correctly Classified Instances	258	98.4733 %
Incorrectly Classified Instances	4	1.5267 %
Kappa statistic	0.9802	
Mean absolute error	0.0224	
Root mean squared error	0.0841	
Relative absolute error	7.2655 %	
Root relative squared error	21.445 %	
Total Number of Instances	262	

=== Confusion Matrix ===

	a	b	c	d	e	<-- classified as
48	0	0	0	0	0	a = drugA
0	34	0	0	1	1	b = drugB
0	0	35	0	0	1	c = drugC
0	0	0	52	1	1	d = drugX
0	1	0	1	89	1	e = drugY

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	drugA
	0,971	0,004	0,971	0,971	0,971	0,967	0,998	0,983	drugB
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	drugC
	0,981	0,005	0,981	0,981	0,981	0,976	1,000	0,999	drugX
	0,978	0,012	0,978	0,978	0,978	0,966	0,998	0,997	drugY
Weighted Avg.	0,985	0,006	0,985	0,985	0,985	0,979	0,999	0,996	

Al observar el área bajo la curva ROC de las distintas clases podemos concluir que el modelo tiene una muy buena capacidad de clasificar correctamente.

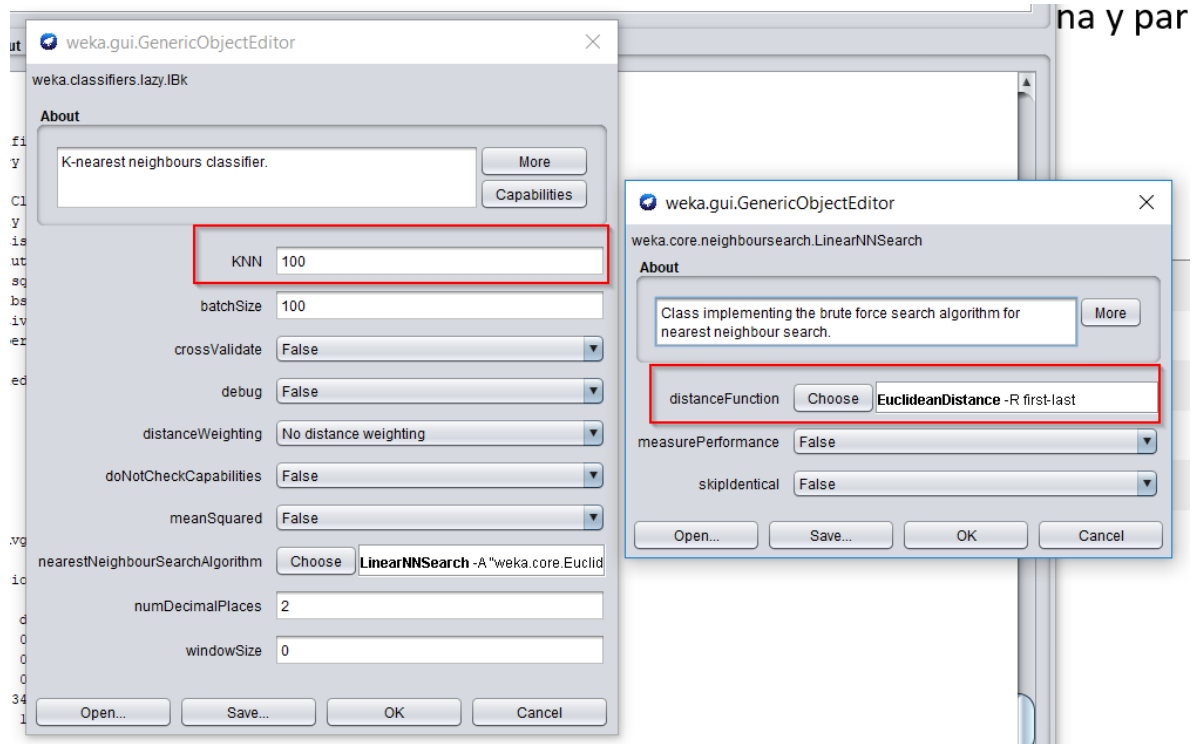
Métodos basados en ejemplos

Se usará el método conocido como k-nearest-neighborh también conocido como **instance-based learning**, aplicaremos el método IBk (Instance Based with K argument) de weka.

Optimizando los parámetros

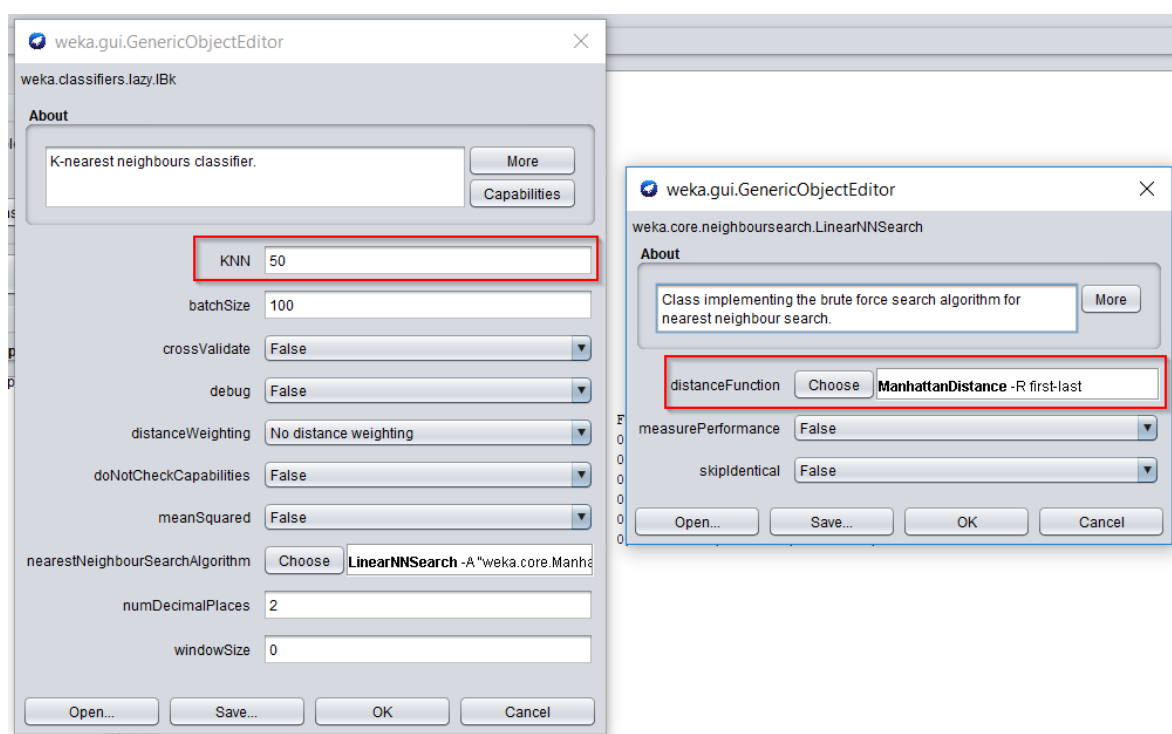
Para obtener mejores resultados vamos a probar el efecto que tiene el parámetro k en la efectividad de este método, haremos 12 pruebas con distintos valores para k y las probaremos con un 10-fold cross validation. Además, observaremos el efecto de la función de distancia en la efectividad, para las 6 primeras pruebas se usará la función de distancia euclidiana y para las restantes 6 se usará la función de distancia manhattan.

Distancia Euclidiana:



VALOR PARA K	EFFECTIVIDAD
1	90.83%
5	84.35%
20	73.66%
50	77.48%
100	62.97%
200	34.73%

Distancia Manhattan:



VALOR PARA K	EFFECTIVIDAD
1	90.83%
5	84.73%
20	78.62%
50	75.57%
100	63.03%
200	34.73%

```

=== Confusion Matrix ===
      a  b  c  d  e  <-- classified as
0  0  0  0  48 |  a = drugA
0  0  0  0  35 |  b = drugB
0  0  0  0  35 |  c = drugC
0  0  0  0  53 |  d = drugX
0  0  0  0  91 |  e = drugY

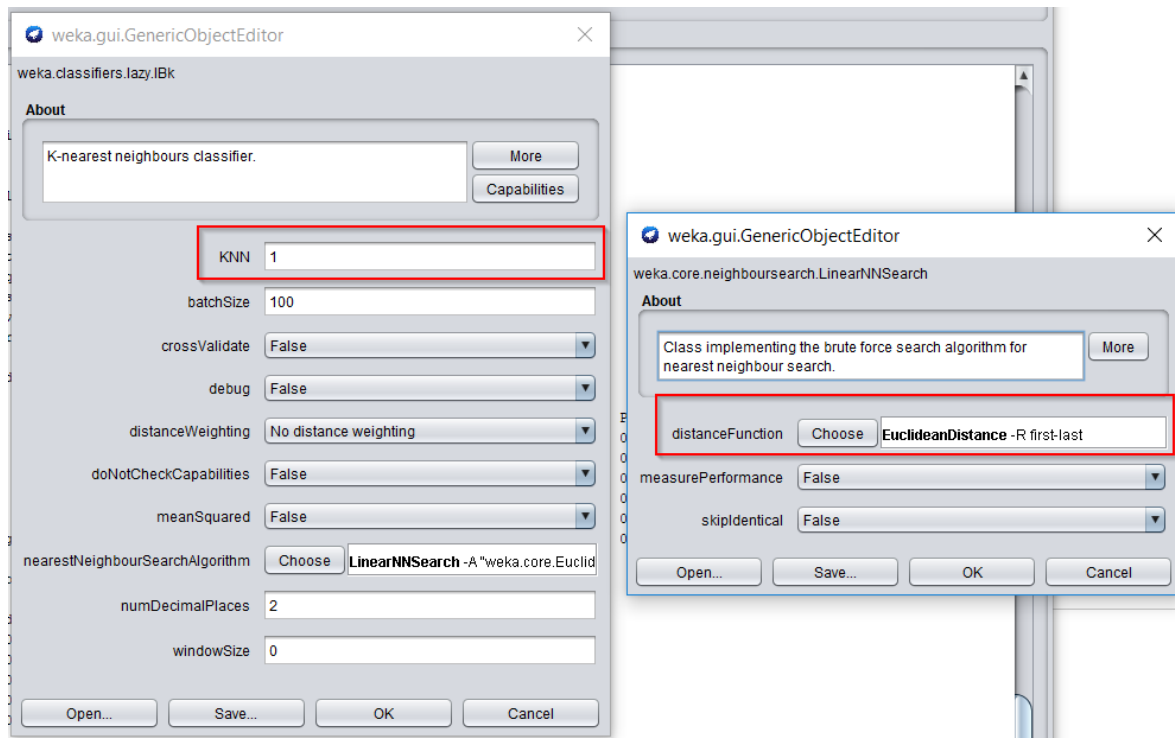
```

Ilustración 2 Matriz de confusión con k=200 y manhattan distance

Después de realizar las pruebas anteriores es posible sacar las siguientes conclusiones:

1. El método IBk funciona mejor cuando la K=1 es decir, cuando la clase de la instancia de ejemplo es la misma que la clase del vecino más cercano.
2. La función de distancia que se utilice implicará cambios menores en la efectividad, que podrían ser despreciables.
3. Cuando k se hace grande respecto al número total de instancias el modelo tiende a predecir que todas las instancias pertenecen a la clase con más instancias en el dataset, esto tiene sentido si estudia la implementación de este método.

Ahora se puede decir con mayor seguridad que los siguientes serian parámetros óptimos para el IBk:



=== Summary ===

Correctly Classified Instances	238	90.8397 %
Incorrectly Classified Instances	24	9.1603 %
Kappa statistic	0.8825	
Mean absolute error	0.0425	
Root mean squared error	0.1896	
Relative absolute error	13.8074 %	
Root relative squared error	48.3309 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,028	0,889	1,000	0,941	0,929	0,985	0,885	drugA
	0,914	0,013	0,914	0,914	0,914	0,901	0,959	0,889	drugB
	1,000	0,018	0,897	1,000	0,946	0,939	0,994	0,929	drugC
	0,962	0,038	0,864	0,962	0,911	0,889	0,954	0,816	drugX
	0,791	0,018	0,960	0,791	0,867	0,815	0,886	0,840	drugY
Weighted Avg.	0,908	0,023	0,913	0,908	0,906	0,879	0,942	0,862	

=== Confusion Matrix ===

a	b	c	d	e	<-- classified as
48	0	0	0	0	a = drugA
2	32	0	0	1	b = drugB
0	0	35	0	0	c = drugC
0	0	0	51	2	d = drugX
4	3	4	8	72	e = drugY

Probando el método con nuevos datos


1	5:drugY	4:drugX	+	0.983
2	5:drugY	5:drugY		0.983
3	5:drugY	4:drugX	+	0.983
4	5:drugY	5:drugY		0.983
5	5:drugY	5:drugY		0.983
6	5:drugY	5:drugY		0.983
7	5:drugY	5:drugY		0.983
8	5:drugY	5:drugY		0.983
9	5:drugY	1:drugA	+	0.983
10	4:drugX	4:drugX		0.983
11	4:drugX	4:drugX		0.983
12	4:drugX	4:drugX		0.983
13	4:drugX	4:drugX		0.983
14	4:drugX	4:drugX		0.983
15	3:drugC	3:drugC		0.983
16	3:drugC	3:drugC		0.983
17	3:drugC	3:drugC		0.983
18	1:drugA	1:drugA		0.983
19	1:drugA	1:drugA		0.983
20	1:drugA	1:drugA		0.983
21	1:drugA	1:drugA		0.983
22	1:drugA	1:drugA		0.983
23	2:drugB	2:drugB		0.983
24	2:drugB	2:drugB		0.983
25	2:drugB	2:drugB		0.983
26	2:drugB	2:drugB		0.983

Máquina de soporte vectorial

Este método es bastante resistente al sobre ajuste, lamentablemente en **Weka** el método `functions.SMO` está restringido a datasets de 2 clases y debido a que nuestro dataset tiene 5 clases se decidió usar el `libSVM` porque este último si soporta datasets con más de 2 clases.

Optimizando los parámetros

Para optimizar los parámetros de este método se va a cambiar el tipo del kernel y se relacionará con el porcentaje de instancias correctamente clasificadas.

 weka.gui.GenericObjectEditor

weka.classifiers.functions.LibSVM

About

A wrapper class for the libsvm library.

More

Capabilities

SVMType

C-SVC (classification)

batchSize

100

cacheSize

40.0

coef0

0.0

cost

1.0

debug

False

degree

3

doNotCheckCapabilities

False

doNotReplaceMissingValues

False

eps

0.001

gamma

0.0

kernelType

radial basis function: $\exp(-\gamma \|u-v\|^2)$

loss

0.1

modelFile

Weka-3-8

normalize

False

nu

0.5

numDecimalPlaces

2

probabilityEstimates

False

seed

1

shrinking

True

weights

Open...

Save...

OK

Cancel

Tipo del Kernel	% Instancias correctamente clasificadas
Lineal	95.03%

Polinomial	41.98%
Radial	85.49%
Sigmoid	71.37%

Después de las pruebas anteriores se puede considerar que los datos de nuestro dataset son linealmente separables debido a que fue este tipo de kernel el que mejor logro clasificar las instancias de ejemplo.

Probando el método con nuevos datos

inst#	actual	predicted	error	prediction
1	5:drugY	4:drugX	+	1
2	5:drugY	5:drugY		1
3	5:drugY	3:drugC	+	1
4	5:drugY	2:drugB	+	1
5	5:drugY	5:drugY		1
6	5:drugY	5:drugY		1
7	5:drugY	5:drugY		1
8	5:drugY	5:drugY		1
9	5:drugY	5:drugY		1
10	5:drugY	5:drugY		1
11	4:drugX	4:drugX		1
12	4:drugX	4:drugX		1
13	4:drugX	4:drugX		1
14	4:drugX	4:drugX		1
15	4:drugX	4:drugX		1
16	3:drugC	3:drugC		1
17	3:drugC	3:drugC		1
18	3:drugC	3:drugC		1
19	1:drugA	1:drugA		1
20	1:drugA	1:drugA		1
21	1:drugA	1:drugA		1
22	1:drugA	1:drugA		1
23	1:drugA	1:drugA		1

Arboles de decisión

Este es un método bueno para problemas donde la clase es nominal, razón por la cual se usará, cabe decir que el output de este algoritmo tiene la gran ventaja de que es muy fácil de entender por las personas lo que facilita su aplicación para futuras instancias de ejemplo.

Este modelo tiene el riesgo de quedar sobreajustado si las ramas son muy profundas.

Vamos a podar el árbol, cabe decir que esto va a significar una menor efectividad con los datos de entrenamiento y prueba, pero nos permitirá obtener mejores resultados para casos desconocidos. Es decir, obtendremos un modelo más general que nos permitirá predecir casos futuros con más precisión.

Optimizando los parámetros

En este caso se observará el árbol y la efectividad del árbol si lo podamos o no lo podamos además se realizará una validación cruzada con un fold de 10.

Árbol Podado:

weka.gui.GenericObjectEditor

weka.classifiers.trees.J48

About

Class for generating a pruned or unpruned C4. [More](#) [Capabilities](#)

batchSize 100

binarySplits False

collapseTree True

confidenceFactor 0.25

debug False

doNotCheckCapabilities False

doNotMakeSplitPointActualValue False

minNumObj 2

numDecimalPlaces 2

numFolds 3

reducedErrorPruning False

saveInstanceData False

seed 1

subtreeRaising True

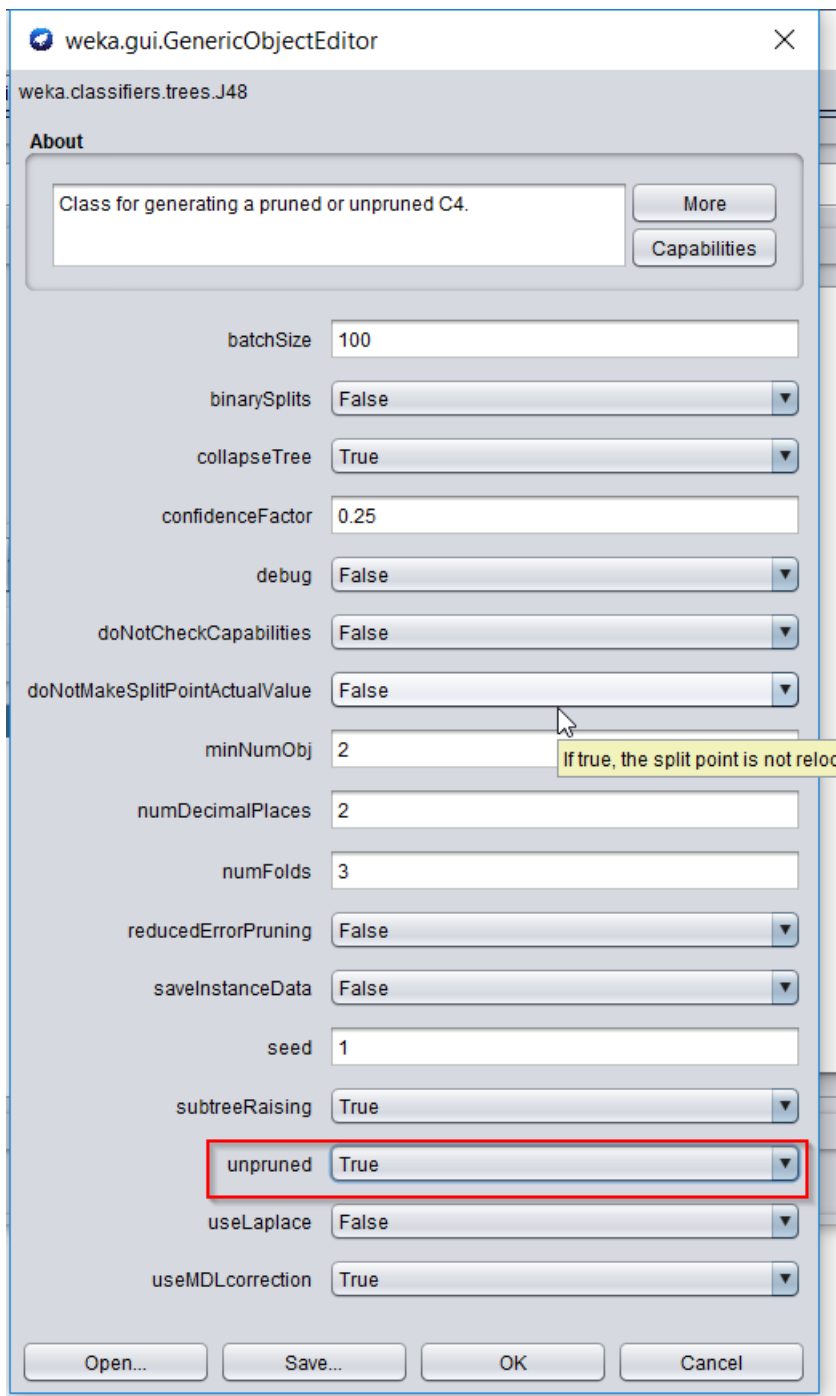
unpruned False

useLaplace False

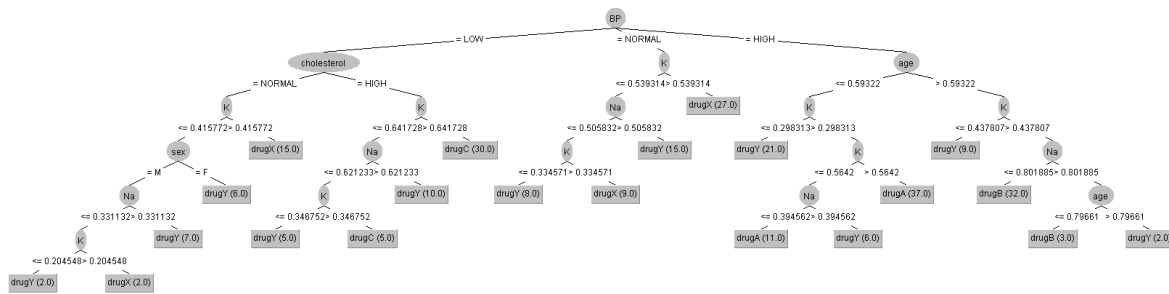
useMDLcorrection True

Open... Save... OK Cancel

Obtenemos un árbol con 20 nodos hoja y 38 nodos en total:



Se obtiene un árbol con 21 nodos hojas y 40 nodos en total (2 más que el árbol podado)



=== Summary ===

Correctly Classified Instances	231	88.1679 %
Incorrectly Classified Instances	31	11.8321 %
Kappa statistic	0.8483	
Mean absolute error	0.0463	
Root mean squared error	0.2037	
Relative absolute error	15.0236 %	
Root relative squared error	51.9142 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,979	0,028	0,887	0,979	0,931	0,916	0,976	0,872	drugA
	0,914	0,018	0,889	0,914	0,901	0,886	0,952	0,875	drugB
	0,943	0,031	0,825	0,943	0,880	0,863	0,975	0,863	drugC
	0,962	0,033	0,879	0,962	0,919	0,899	0,982	0,923	drugX
	0,747	0,041	0,907	0,747	0,819	0,744	0,901	0,838	drugY
Weighted Avg.	0,882	0,033	0,884	0,882	0,879	0,842	0,948	0,870	

=== Confusion Matrix ===

a	b	c	d	e	<-- classified as
47	0	0	0	1	a = drugA
1	32	0	0	2	b = drugB
0	0	33	0	2	c = drugC
0	0	0	51	2	d = drugX
5	4	7	7	68	e = drugY

Con los cambios realizados anteriormente no se encontró una mejoría significativa en la efectividad del modelo, sin embargo, se llegó a un árbol 2 nodos más simples.

Debido a este poco cambio se decidió probar variaciones en el parámetro minNumObj el cual determina el número mínimo de instancias que puede haber en un nodo hoja, aumentar este valor disminuirá el número de nodos hojas en el árbol y por lo tanto ayuda a simplificarlo más.

weka.gui.GenericObjectEditor

weka.classifiers.trees.J48

About

Class for generating a pruned or unpruned C4. More Capabilities

batchSize 100

binarySplits False

collapseTree True

confidenceFactor 0.25

debug False

doNotCheckCapabilities False

doNotMakeSplitPointActualValue False

minNumObj 32

numDecimalPlaces 2

numFolds 3

reducedErrorPruning False

saveInstanceData False

seed 1

subtreeRaising True

unpruned False

useLaplace False

useMDLcorrection True

Open... Save... OK Cancel

minNumObj	Total nodes	Leaf nodes	Correctly Clasified Instances
1	38	20	90.83%
2	38	20	89.31%
4	30	16	91.22%
8	26	14	85.87%
16	18	10	78.24%
24	14	8	71.75%

The decision tree starts with a root node 'BP'. It branches into three main categories: '= LOW', '= NORMAL', and '= HIGH'. The '= LOW' branch leads to a node 'cholesterol', which further branches into '= NORMAL' and '= HIGH'. The '= NORMAL' branch leads to a node 'K', which then branches into two leaf nodes: 'drugY (17.0/2.0)' and 'drugX (15.0)'. The '= HIGH' branch leads to a node 'K', which then branches into two leaf nodes: 'Na' and 'drugC (30.0)'. The '= NORMAL' branch leads to a node 'K', which then branches into two leaf nodes: 'Na' and 'drugX (27.0)'. The '= HIGH' branch leads to a node 'age', which then branches into two leaf nodes: 'Na' and 'drugX (9.0)'. The '= NORMAL' branch leads to a node 'K', which then branches into two leaf nodes: 'drugY (21.0)' and 'Na'. The '= HIGH' branch leads to a node 'K', which then branches into two leaf nodes: 'drugY (9.0)' and 'drugB (37.0/2.0)'. The decision tree is a hierarchical structure that uses decision rules to recommend a drug based on the input features.

Probando el método con nuevos datos

inst#	actual	predicted	error	prediction
1	5:drugY	5:drugY	1	
2	5:drugY	5:drugY	1	
3	5:drugY	3:drugC	+	1
4	5:drugY	2:drugB	+	0.969
5	5:drugY	5:drugY	1	
6	5:drugY	5:drugY	1	
7	5:drugY	5:drugY	1	
8	5:drugY	5:drugY	1	
9	5:drugY	5:drugY	0.875	
10	5:drugY	5:drugY	1	
11	4:drugX	4:drugX	1	
12	4:drugX	4:drugX	1	
13	4:drugX	4:drugX	1	
14	4:drugX	4:drugX	1	
15	4:drugX	4:drugX	1	
16	3:drugC	3:drugC	1	
17	3:drugC	3:drugC	1	
18	3:drugC	3:drugC	1	
19	1:drugA	1:drugA	1	
20	1:drugA	1:drugA	1	
21	1:drugA	1:drugA	1	
22	1:drugA	1:drugA	1	
23	1:drugA	1:drugA	1	
24	2:drugB	2:drugB	0.969	

Bonus

Se probó rápidamente el método Random Forest para probar su efectividad.

=== Summary ===

Correctly Classified Instances	246	93.8931 %
Incorrectly Classified Instances	16	6.1069 %
Kappa statistic	0.921	
Mean absolute error	0.0668	
Root mean squared error	0.1483	
Relative absolute error	21.6939 %	
Root relative squared error	37.8053 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

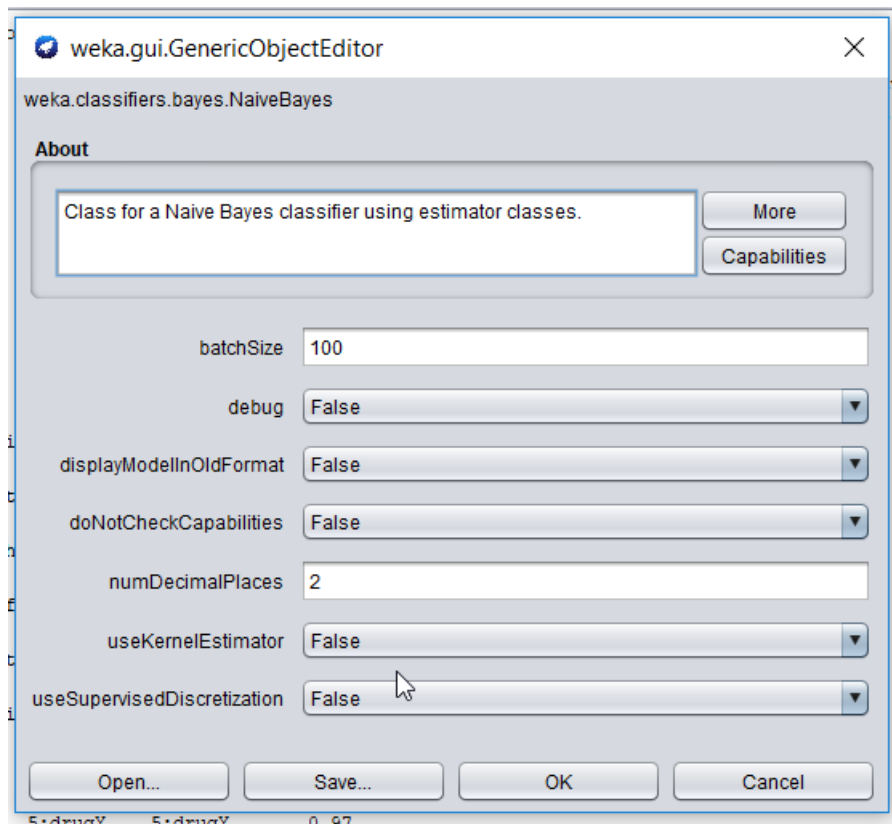
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,023	0,906	1,000	0,950	0,940	0,998	0,991	drugA
	0,971	0,004	0,971	0,971	0,971	0,967	0,999	0,991	drugB
	0,971	0,009	0,944	0,971	0,958	0,951	0,997	0,978	drugC
	0,925	0,014	0,942	0,925	0,933	0,917	0,997	0,988	drugX
	0,890	0,029	0,942	0,890	0,915	0,873	0,985	0,976	drugY
Weighted Avg.	0,939	0,019	0,940	0,939	0,939	0,917	0,993	0,983	

=== Confusion Matrix ===

a	b	c	d	e	<-- classified as
48	0	0	0	0	a = drugA
1	34	0	0	0	b = drugB
0	0	34	0	1	c = drugC
0	0	0	49	4	d = drugX
4	1	2	3	81	e = drugY

Método Bayesiano

Se usará a continuación el método bayesiano ingenuo el cual es un clasificador probabilístico que se fundamenta en el teorema de Bayes de probabilidad condicional.



Debido a que este método se fundamenta casi que completamente en el teorema de Bayes no hay muchos parámetros que podamos configurar y optimizar.

Se probó el modelo mediante una validación cruzada con un **fold** de 10

=== Summary ===

Correctly Classified Instances	237	90.458 %
Incorrectly Classified Instances	25	9.542 %
Kappa statistic	0.8758	
Mean absolute error	0.0714	
Root mean squared error	0.1744	
Relative absolute error	23.1827 %	
Root relative squared error	44.4498 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,896	0,019	0,915	0,896	0,905	0,884	0,994	0,976	drugA
	0,971	0,013	0,919	0,971	0,944	0,936	0,997	0,983	drugB
	0,943	0,013	0,917	0,943	0,930	0,919	0,989	0,958	drugC
	0,868	0,010	0,958	0,868	0,911	0,891	0,993	0,980	drugX
	0,890	0,076	0,862	0,890	0,876	0,808	0,976	0,961	drugY
Weighted Avg.	0,905	0,035	0,906	0,905	0,905	0,871	0,987	0,970	

=== Confusion Matrix ===

```
a b c d e <-- classified as
43 0 0 0 5 | a = drugA
1 34 0 0 0 | b = drugB
0 0 33 0 2 | c = drugC
0 1 0 46 6 | d = drugX
3 2 3 2 81 | e = drugY
```

El área debajo de la curva ROC y el porcentaje instancias correctamente clasificados muestran que en términos generales este método funciona bien.

Probando el método con datos nuevos

inst#	actual	predicted	error	prediction
1	5:drugY	5:drugY		0.551
2	5:drugY	5:drugY		0.996
3	5:drugY	3:drugC	+	0.83
4	5:drugY	2:drugB	+	0.925
5	5:drugY	5:drugY		0.843
6	5:drugY	5:drugY		0.993
7	5:drugY	5:drugY		0.945
8	5:drugY	5:drugY		1
9	5:drugY	5:drugY		0.997
10	5:drugY	5:drugY		0.988
11	4:drugX	4:drugX		0.826
12	4:drugX	4:drugX		0.644
13	4:drugX	4:drugX		0.666
14	4:drugX	4:drugX		0.974
15	4:drugX	4:drugX		0.647
16	3:drugC	3:drugC		0.876
17	3:drugC	3:drugC		0.73
18	3:drugC	3:drugC		0.785
19	1:drugA	1:drugA		0.979
20	1:drugA	1:drugA		0.949
21	1:drugA	1:drugA		0.984
22	1:drugA	5:drugY	+	0.708
23	1:drugA	1:drugA		0.982
24	2:drugB	2:drugB		0.965
25	2:drugB	2:drugB		0.949
26	2:drugB	2:drugB		0.929
27	2:drugB	2:drugB		0.619

Redes neuronales

Debido a que en el problema de clasificación que estamos trabajando hay más de 2 clases se procederá a usar un perceptrón multicapa dado que el perceptrón simple tan solo discrimina entre dos clases linealmente separables.

hiddenLayers: neuronas por capa (2,3)

'a' = (attribs + classes) / 2,

'i' = attribs,

'o' = classes

t' = attribs + classes

A diferencia del método discutido anteriormente este si tiene varios parámetros para modificar y probar, para probar el modelo usaremos una validación cruzada con 10 folds, lo primero que haremos será probar varios valores para el numero de capas internas y tamaño de estas.

weka.gui.GenericObjectEditor

weka.classifiers.functions.MultilayerPerceptron

About

A Classifier that uses backpropagation to classify instances. [More](#) [Capabilities](#)

GUI

autoBuild

batchSize

debug

decay

doNotCheckCapabilities

hiddenLayers

learningRate This defines the hidden layers of the neural network

momentum

nominalToBinaryFilter

normalizeAttributes

normalizeNumericClass

numDecimalPlaces

reset

seed

trainingTime

validationSetSize

validationThreshold

hiddenLayers = 'a'

=== Summary ===

Correctly Classified Instances	259	98.855 %
Incorrectly Classified Instances	3	1.145 %
Kappa statistic	0.9851	
Mean absolute error	0.0174	
Root mean squared error	0.0707	
Relative absolute error	5.6478 %	
Root relative squared error	18.0251 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	drugA
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	drugB
	0,971	0,004	0,971	0,971	0,971	0,967	1,000	0,999	drugC
	0,981	0,000	1,000	0,981	0,990	0,988	0,999	0,998	drugX
	0,989	0,012	0,978	0,989	0,984	0,975	0,999	0,998	drugY
Weighted Avg.	0,989	0,005	0,989	0,989	0,989	0,984	0,999	0,999	

=== Confusion Matrix ===

```
a b c d e <-- classified as
48 0 0 0 0 | a = drugA
0 35 0 0 0 | b = drugB
0 0 34 0 1 | c = drugC
0 0 0 52 1 | d = drugX
0 0 1 0 90 | e = drugY
```

hiddenLayers = 2,5

=== Summary ===

Correctly Classified Instances	213	81.2977 %
Incorrectly Classified Instances	49	18.7023 %
Kappa statistic	0.7544	
Mean absolute error	0.0956	
Root mean squared error	0.2245	
Relative absolute error	31.034 %	
Root relative squared error	57.2373 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,979	0,159	0,580	0,979	0,729	0,687	0,917	0,609	drugA
	0,000	0,000	?	0,000	?	?	0,882	0,414	drugB
	0,914	0,026	0,842	0,914	0,877	0,858	0,958	0,895	drugC
	0,868	0,014	0,939	0,868	0,902	0,879	0,993	0,976	drugX
	0,967	0,035	0,936	0,967	0,951	0,925	0,996	0,991	drugY
Weighted Avg.	0,813	0,048	?	0,813	?	?	0,961	0,828	

=== Confusion Matrix ===

```
a b c d e <-- classified as
47 0 1 0 0 | a = drugA
33 0 0 0 2 | b = drugB
0 0 32 1 2 | c = drugC
0 0 5 46 2 | d = drugX
1 0 0 2 88 | e = drugY
```

hiddenLayers = 5, 7

=== Summary ===

Correctly Classified Instances	257	98.0916 %
Incorrectly Classified Instances	5	1.9084 %
Kappa statistic	0.9752	
Mean absolute error	0.0202	
Root mean squared error	0.0771	
Relative absolute error	6.5439 %	
Root relative squared error	19.6482 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,005	0,980	1,000	0,990	0,987	1,000	1,000	drugA
	0,943	0,000	1,000	0,943	0,971	0,967	1,000	1,000	drugB
	1,000	0,009	0,946	1,000	0,972	0,968	1,000	1,000	drugC
	0,943	0,000	1,000	0,943	0,971	0,964	1,000	0,999	drugX
	1,000	0,012	0,978	1,000	0,989	0,983	1,000	1,000	drugY
Weighted Avg.	0,981	0,006	0,982	0,981	0,981	0,976	1,000	1,000	

=== Confusion Matrix ===

```
a b c d e <-- classified as
48 0 0 0 0 | a = drugA
1 33 0 0 1 | b = drugB
0 0 35 0 0 | c = drugC
0 0 2 50 1 | d = drugX
0 0 0 0 91 | e = drugY
```


hiddenLayers = 5,5

=== Summary ===

Correctly Classified Instances	259
Incorrectly Classified Instances	3
Kappa statistic	0.9851
Mean absolute error	0.0166
Root mean squared error	0.0614
Relative absolute error	5.386 %
Root relative squared error	15.6525 %
Total Number of Instances	262

98.855 %
1.145 %

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,979	0,005	0,979	0,979	0,979	0,974	1,000	0,999	drugA
	0,971	0,004	0,971	0,971	0,971	0,967	1,000	0,998	drugB
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	drugC
	0,981	0,000	1,000	0,981	0,990	0,988	1,000	1,000	drugX
	1,000	0,006	0,989	1,000	0,995	0,992	1,000	1,000	drugY
Weighted Avg.	0,989	0,003	0,989	0,989	0,989	0,986	1,000	1,000	

=== Confusion Matrix ===

	a	b	c	d	e	<-- classified as
47	1	0	0	0	0	a = drugA
1	34	0	0	0	0	b = drugB
0	0	35	0	0	0	c = drugC
0	0	0	52	1	0	d = drugX
0	0	0	0	91	0	e = drugY

hiddenLayers = 7,5

=== Summary ===

Correctly Classified Instances	260	99.2366 %
Incorrectly Classified Instances	2	0.7634 %
Kappa statistic	0.9901	
Mean absolute error	0.0176	
Root mean squared error	0.0614	
Relative absolute error	5.7247 %	
Root relative squared error	15.6506 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	drugA
	0,971	0,000	1,000	0,971	0,986	0,983	1,000	1,000	drugB
	1,000	0,004	0,972	1,000	0,986	0,984	1,000	0,999	drugC
	0,981	0,000	1,000	0,981	0,990	0,988	1,000	0,999	drugX
	1,000	0,006	0,989	1,000	0,995	0,992	1,000	1,000	drugY
Weighted Avg.	0,992	0,003	0,993	0,992	0,992	0,990	1,000	1,000	

=== Confusion Matrix ===

```
a b c d e <-- classified as
48 0 0 0 0 | a = drugA
0 34 0 0 1 | b = drugB
0 0 35 0 0 | c = drugC
0 0 1 52 0 | d = drugX
0 0 0 0 91 | e = drugY
```

hiddenLayers = 7,7

=== Summary ===

Correctly Classified Instances	257	98.0916 %
Incorrectly Classified Instances	5	1.9084 %
Kappa statistic	0.9752	
Mean absolute error	0.0177	
Root mean squared error	0.0733	
Relative absolute error	5.7626 %	
Root relative squared error	18.6772 %	
Total Number of Instances	262	

=== Detailed Accuracy By Class ===

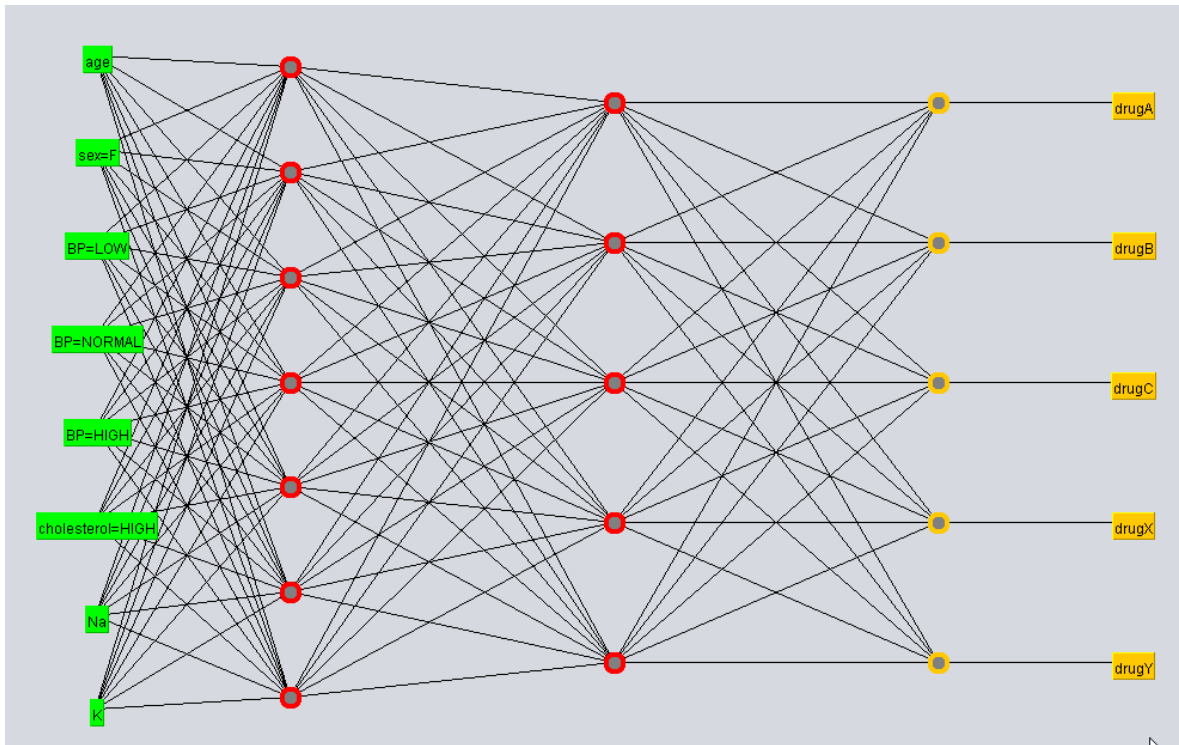
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,005	0,980	1,000	0,990	0,987	1,000	1,000	drugA
	0,971	0,000	1,000	0,971	0,986	0,983	1,000	0,999	drugB
	1,000	0,009	0,946	1,000	0,972	0,968	1,000	1,000	drugC
	0,943	0,000	1,000	0,943	0,971	0,964	1,000	0,998	drugX
	0,989	0,012	0,978	0,989	0,984	0,975	0,999	0,999	drugY
Weighted Avg.	0,981	0,006	0,981	0,981	0,981	0,975	1,000	0,999	

=== Confusion Matrix ===

```
a b c d e <-- classified as
48 0 0 0 0 | a = drugA
1 34 0 0 0 | b = drugB
0 0 35 0 0 | c = drugC
0 0 1 50 2 | d = drugX
0 0 1 0 90 | e = drugY
```

Observaciones

Luego de probar con varios tamaños para una red neuronal de 2 capas ocultas se encontró un tamaño para las capas que logro una efectividad del 99%



El tamaño de la primera capa es igual al número de atributos de cada instancia (incluyendo el atributo clase) y el tamaño de la segunda capa es el número de valores que tiene la clase.

Se observó además que en la capa de entrada no están todos los atributos de las instancias, a continuación, se detallan los pesos calculados para cada nodo.

Sigmoid Node 0

Inputs	Weights
Threshold	-4.1183471546270765
Node 12	-1.5116455534001523
Node 13	-3.9286113086610372
Node 14	8.900718644755173
Node 15	-3.96805826251309
Node 16	-8.926891275335846

Sigmoid Node 1

Inputs	Weights
Threshold	-11.083912574406314
Node 12	-2.357279431246721
Node 13	-3.5170522387865892
Node 14	7.670026623993085
Node 15	-3.600669824216526
Node 16	7.572690633088281

Sigmoid Node 2

Inputs	Weights
Threshold	4.297626515883453
Node 12	-2.569330640774673
Node 13	-1.679265111536604
Node 14	-8.084277600309134
Node 15	-1.7592586274415312
Node 16	-8.856502160162846

Sigmoid Node 3

Inputs	Weights
Threshold	-3.6902719972255893
Node 12	-2.240145999625349
Node 13	-0.9994806984177441
Node 14	-9.977963526359714
Node 15	-0.4413401209463021
Node 16	8.539069544538611

Sigmoid Node 4

Inputs	Weights
Threshold	-8.812907917825582
Node 12	4.807566753811246
Node 13	5.5106115476854285
Node 14	2.0132595414171184
Node 15	5.061956798003515
Node 16	0.7415272557399901

--

Sigmoid Node 5

Inputs	Weights
Threshold	-1.595198772907888
Attrib age	-0.8038201286636749
Attrib sex=F	-0.22946927816683357
Attrib BP=LOW	0.6083488080492804
Attrib BP=NORMAL	-2.8708588661163397
Attrib BP=HIGH	3.85101677924977
Attrib cholesterol=HIGH	3.9322817729121784
Attrib Na	-0.8129544851863341
Attrib K	3.3392782852349217

Sigmoid Node 6

Inputs	Weights
Threshold	0.7647086664186084
Attrib age	0.07482984706139284
Attrib sex=F	0.21818762052388618
Attrib BP=LOW	-0.4025901208117985
Attrib BP=NORMAL	-0.21438280426296155
Attrib BP=HIGH	-0.22050928387510677
Attrib cholesterol=HIGH	0.03837677858395895
Attrib Na	-3.826313179737443
Attrib K	7.411139063016201

Sigmoid Node 7

Inputs	Weights
Threshold	-1.9218415470941377
Attrib age	-3.1107381156773544
Attrib sex=F	-0.17471607685508156
Attrib BP=LOW	1.6121326783204843
Attrib BP=NORMAL	-1.358953917868424
Attrib BP=HIGH	1.660401069878046
Attrib cholesterol=HIGH	0.4648445013899239
Attrib Na	-1.5218497869883059
Attrib K	3.1753603436253948

Sigmoid Node 8

Inputs	Weights
Threshold	-2.2741666195711243
Attrib age	-0.11919900003142211
Attrib sex=F	0.46313019305443814
Attrib BP=LOW	0.4489959308865731
Attrib BP=NORMAL	-1.7075039244351182
Attrib BP=HIGH	3.573486953385733
Attrib cholesterol=HIGH	2.6566062845628693
Attrib Na	-1.5590973627753872
Attrib K	5.063048366452948

Sigmoid Node 9

Inputs	Weights
Threshold	0.691654504849297
Attrib age	0.2940518761979899
Attrib sex=F	-0.03498581138365741
Attrib BP=LOW	-0.27051264825170745
Attrib BP=NORMAL	-0.3048863306780871
Attrib BP=HIGH	-0.18652377911553952
Attrib cholesterol=HIGH	-0.15174163164825383
Attrib Na	-3.70825310004504
Attrib K	7.32434685785575

Sigmoid Node 10

Inputs	Weights
Threshold	0.37606068719678537
Attrib age	-9.045368350239896
Attrib sex=F	-0.47388227329811966
Attrib BP=LOW	3.4872823837377815
Attrib BP=NORMAL	-3.906511358042003
Attrib BP=HIGH	0.0385477535111464
Attrib cholesterol=HIGH	-0.006073927539852128
Attrib Na	-1.8053604515803539
Attrib K	1.6181389405623547

Sigmoid Node 11

Inputs	Weights
Threshold	-2.789154381067088
Attrib age	-0.2706335879034606
Attrib sex=F	0.20205504980409572
Attrib BP=LOW	3.326006010767991
Attrib BP=NORMAL	2.9546988714736124
Attrib BP=HIGH	-3.500333882018348
Attrib cholesterol=HIGH	-0.2831110109617154
Attrib Na	-3.8160234615366653
Attrib K	7.650512434858828

PESOS NODOS CAPA DE SALIDA

Sigmoid Node 12

Inputs	Weights
Threshold	3.3238331094009776
Node 5	-0.07555228230010039
Node 6	-2.74810579652247
Node 7	0.14662702697979574
Node 8	-1.2850476285951813
Node 9	-2.7377646512195706
Node 10	1.6921103044644157
Node 11	-3.4330217419458022

Sigmoid Node 13

Inputs	Weights
Threshold	4.572726442005464
Node 5	-0.2434732331744647
Node 6	-3.673048984743559
Node 7	-0.7529981485333217
Node 8	-2.025294238030296
Node 9	-3.3875270773190382
Node 10	1.1631097508159394
Node 11	-2.1996802885585285

Sigmoid Node 14

Inputs	Weights
Threshold	2.6086796213788053
Node 5	1.665269661668782
Node 6	-0.5242842245601238
Node 7	0.12553546258475576
Node 8	2.119425352264825
Node 9	-0.9221181009460448
Node 10	-0.38382585223228616
Node 11	-8.009108764455606

Sigmoid Node 15

Inputs	Weights
Threshold	4.557844546883982
Node 5	-0.5640302761640077
Node 6	-3.1883761426819732
Node 7	-0.7914061291643032
Node 8	-2.2088641732946397
Node 9	-2.975798716370475
Node 10	1.1116127925514085
Node 11	-1.5269581928732592

Sigmoid Node 16

Inputs	Weights
Threshold	7.939418035369441
Node 5	-5.484941275063864
Node 6	2.5982694859484234
Node 7	-2.418949616157796
Node 8	-2.9873684552341353
Node 9	3.179948922473734
Node 10	-8.249359096849041
Node 11	-0.6169907938593906

Probando el método con datos nuevos

inst#	actual	predicted	error	predicti
1	5:drugY	5:drugY	0.972	
2	5:drugY	5:drugY	0.992	
3	5:drugY	5:drugY	0.803	
4	5:drugY	5:drugY	0.745	
5	5:drugY	5:drugY	0.998	
6	5:drugY	5:drugY	0.999	
7	5:drugY	5:drugY	0.999	
8	5:drugY	5:drugY	0.999	
9	5:drugY	5:drugY	0.999	
10	5:drugY	5:drugY	0.999	
11	4:drugX	4:drugX	0.984	
12	4:drugX	4:drugX	0.985	
13	4:drugX	4:drugX	0.985	
14	4:drugX	4:drugX	0.984	
15	4:drugX	4:drugX	0.984	
16	3:drugC	3:drugC	0.964	
17	3:drugC	3:drugC	0.981	
18	3:drugC	3:drugC	0.982	
19	1:drugA	1:drugA	0.971	
20	1:drugA	1:drugA	0.969	
21	1:drugA	1:drugA	0.955	
22	1:drugA	1:drugA	0.976	
23	1:drugA	1:drugA	0.972	
24	2:drugB	2:drugB	0.982	
25	2:drugB	2:drugB	0.978	
26	2:drugB	2:drugB	0.976	
27	2:drugB	2:drugB	0.831	
1	5:drugY	5:drugY	0.993	
2	5:drugY	5:drugY	0.998	
3	5:drugY	5:drugY	0.998	
4	5:drugY	5:drugY	0.994	
5	5:drugY	5:drugY	0.995	
6	5:drugY	5:drugY	0.985	
7	5:drugY	5:drugY	0.993	
8	5:drugY	5:drugY	0.99	
9	5:drugY	5:drugY	0.995	

Conclusiones, Eligiendo a los ganadores

A continuación, se compacta en una tabla el porcentaje de instancias correctamente clasificadas por cada método de aprendizaje.

Método Aprendizaje	% Instancias correctamente clasificadas
Regresión logística	98.00%
K-Neighbors	90.83%
Máquina de soporte vectorial	95.03%
Árbol de decisión	90.83%
Método Bayesiano	90.45%
Red Neuronal	99.23%

Antes de elegir a un ganador es importante aclarar que no es conveniente basarse únicamente en la efectividad clasificando las instancias de prueba, también se debe tener en cuenta las instancias

futuras que se desean predecir, por lo tanto, la posibilidad de sobreajuste es un factor que hay que tener muy en cuenta. Además, también se deben considerar factores subjetivos en el modelo resultante como por ejemplo la facilidad de ser entendido por un humano de tal forma que sea intuitivo para un humano sacar conocimiento valioso de forma eficaz.

Dicho lo anterior la medalla de oro se la lleva la máquina de soporte vectorial por su relativamente alta efectividad y poca posibilidad de sobreajuste debido a que se basa en pocas instancias del dataset para hallar el modelo.

La medalla de plata se la lleva el árbol de decisión porque el modelo resultante tiene una buena efectividad y es muy fácil de entender por humanos, lo cual lo hace ideal para sacar conocimiento intuitivo de forma rápida y eficaz.

La medalla de bronce se la llevan la regresión logística y la red neuronal por su alta efectividad, sin embargo de la red neuronal quedamos con el temor de que el modelo este sobreajustado, la única forma de averiguarlo es con datos de prueba nuevos.