WEKA

PREPARACIÓN DE DATOS

Preparando archivo:

Ponemos un título a cada atributo (columna) del archivo de datos ***Drug.csv***

```
1 Age,Sex,Blood_Pressure,Cholesterol,Na,K,Drug
2 23,F,HIGH,HIGH,0.792535,0.031258,drugY
3 47,M,LOW,HIGH,0.739309,0.056468,drugC
```

El resultado se encuentra en el archivo ***Drug_prepared.csv***

Limpieza de datos | datos ausentes/nulos y atípicos:

Se identifican los datos ***ausentes/nulos*** y ***atípicos*** del ***dataset*** usando el filtro ***unsupervized/attribute/NumericalCleaner***.

Primeramente, en el atributo **edad**, con índice 1 y finalmente en el atributo ***Na*** con índice 5.

La siguiente configuración permitió marcar ***edades*** atípicas en el dataset.

- attributeIndices: 1 // apply to attribute with index 1 (Age)
- maxThreshoold: 110 // max age allowed
- maxDefault: NaN
- minThreshoold: 0 // min age allowed
- minDefault: NaN

Se detectó un dato atípico representando el 1% de los datos (edad de 145 años).

La siguiente configuración permitió marcar niveles de **Na** atípicos en el dataset.

- attributeIndices: 5 // apply to attribute with index 5 (Na)
- maxThreshoold: 1 // max Na allowed
- maxDefault: NaN
- minThreshoold: 0 // min Na allowed
- minDefault: NaN

Finalmente, con ayuda del filtro ***unsupervized/instance/RemoveWithValues*** *se eliminaron las instancias con atributos atípicos que fueron detectados.*

El resultado de esta limpieza se encuentra en el archivo ***weka/1_drugs_numeric_cleaned.arff***

Limpieza de datos | Transformación de los datos:

Los datos fueron suavizados utilizando el filtro ***supervized/instance/SMOTE****, el resultado de esta transformación se encuentra en el archivo* ***weka/2_drugs_smoothed.arff***

Luego, Los atributos numéricos fueron normalizados utilizando el filtro ***unsupervised/attribute/Normalize****, el resultado de esta transformación se encuentra en el archivo* ***weka/2_drugs_normalized.arff***

Finalmente, los atributos ***Na*** y ***K*** fueron discretizados a 6 posibles valores utilizando el filtro ***unsupervised/attribute/discretize***, el resultado de esta transformación se encuentra en el archivo ***weka/3_drugs_discretized.arff***

Limpieza de datos | Otras notas**:**

Se optó por no eliminar variables del dataset, pues se hicieron pruebas donde se eliminaban algunos atributos y los resultados nunca fueron mejores a aquellos producidos con el dataset completo.

APLICANDO METODOS

1_drugs_numeric_cleaned.arff | Árbol de decisión (trees/J48)

ConfidenceFactor	minNumObj	Correct	Incorrect
0.4	1	175 (88.38 %)	23 (11.61 %)
0.25	2	173 (87.37 %)	25 (12.62 %)
0.1	1	176 (88.88 %)	22 (11.11 %)
0.05	1	174 (87.87 %)	24 (12.12 %)
0.01	1	173 (87.37 %)	25 (12.62 %)

Cuando ***confidenceFactor*** es igual a 0.1:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.942	0.931	0.930	0.969	0.926

1_drugs_numeric_cleaned.arff | Métodos bayesianos (bayes/NaiveBayes)

Correct	Incorrect
175 (88.38 %)	23 (11.61 %)

	drugA	drugB	drugC	drugX	drugY
ROC area	0.985	0.997	0.972	0.984	0.970

1_drugs_numeric_cleaned.arff | support vector machine (functions/SMO)

Solo útil para predecir dos clases, por tanto, no es funcional en este caso donde estamos intentando predecir cinco.

Complexity	Correct	Incorrect
60.0	190 (95.95 %)	8 (4.04%)
30.0	189 (95.45 %)	9 (4.54 %)
2.5	185 (93.43 %)	13 (6.56 %)
2.0	183 (92.42 %)	15 (7.57 %)
1.5	181 (91.41 %)	17 (8.58 %)

1.0	183 (92.42 %)	15 (7.57 %)
0.5	172 (86.86 %)	26 (13.13%)
0.0	97 (45.95 %)	107 (54.04%)

Cuando ***complexity*** es igual a 60.0:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.997	0.984	0.991	1.0	0.993

1_drugs_numeric_cleaned.arff **|** support vector machine (functions/libSVM)

Cost	Correct	Incorrect
2.5	88 (44.94 %)	109 (55.05 %)
2.0	82 (42.42 %)	114 (57.57 %)
1.5	79 (39.89 %)	109 (60.10 %)
1.0	84 (40.90 %)	117 (59.09 %)
0.5	90 (45.45 %)	108 (13.13%)

Cuando ***cost*** es igual a 0.5:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.5	0.5	0.497	0.5	0.495

1_drugs_numeric_cleaned.arff **|** Neural networks (functions/MultilayerPerceptron)

learningRate	Correct	Incorrect
0.4	194 (97.97 %)	4 (2.02 %)
0.3	195 (98.48 %)	3 (1.51 %)
0.2	194 (97.97 %)	4 (2.02 %)

Cuando ***learningRate*** es igual a 0.3:

	drugA	drugB	drugC	drugX	drugY
ROC area	1.0	1.0	1.0	1.0	0.999

1_drugs_numeric_cleaned.arff **|** Método basado en ejemplos (lazy/IBK)

distanceFunction	Correct	Incorrect
Chebyshev	162 (81.81 %)	36 (18.81 %)
Euclidean	167 (84.34 %)	31 (15.65 %)
Filtered	110 (55.55 %)	88 (44.44 %)
Manhattan	170 (85.85 %)	28 (14.14 %)
Minkowski	167 (84.34 %)	31 (15.65 %)

Cuando ***distanceFunction*** es Manhattan:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.917	0.934	0.968	0.920	0.876

1_drugs_numeric_cleaned.arff | Regression logística (functions/SimpleLogistic)

Correct	Incorrect
191 (96.46 %)	7 (3.53 %)

	drugA	drugB	drugC	drugX	drugY
ROC area	1.0	0.997	1.0	0.999	0.997

2_drugs_smoothed/normalized.arff | Árbol de decisión (trees/J48)

ConfidenceFactor	minNumObj	Correct	Incorrect
0.4	1	198 (92.52 %)	16 (7.47 %)
0.25	2	195 (91.12 %)	19 (8.87 %)
0.1	1	199 (92.99 %)	15 (7.00 %)
0.05	1	198 (92.52 %)	16 (7.47 %)
0.01	1	196 (91.58 %)	18 (8.41 %)

ando ***confidenceFactor*** es igual a 0.1:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.947	0.967	0.964	0.958	0.949

2_drugs_smoothed/normalized.arff | Métodos bayesianos (bayes/NaiveBayes)

Correct	Incorrect
190 (88.78 %)	24 (11.21 %)

	drugA	drugB	drugC	drugX	drugY
ROC area	0.986	0.996	0.989	0.989	0.971

2_drugs_smoothed/normalized.arff **|** support vector machine (functions/SMO)

Complexity	Correct	Incorrect
60.0	205 (95.79 %)	9 (4.20 %)
30.0	205 (95.79 %)	9 (4.20 %)
2.5	199 (92.99 %)	15 (7.00 %)
2.0	197 (92.05 %)	17 (7.94 %)
1.5	195 (91.12 %)	19 (8.87 %)
1.0	192 (89.71 %)	22 (10.28 %)
0.5	185 (86.44 %)	29 (13.55 %)
0.0	91 (42.52 %)	123 (57.47%)

Cuando ***complexity*** es igual a 60.0:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.997	0.974	0.997	0.993	0.994

2_drugs_smoothed/normalized.arff **|** support vector machine (functions/libSVM)

Cost	Correct	Incorrect
2.5	96 (44.85 %)	118 (55.14 %)
2.0	94 (43.92 %)	120 (56.07 %)
1.5	87 (40.65 %)	127 (59.34 %)
1.0	90 (40.05 %)	124 (57.94 %)
0.5	91 (42.52 %)	123 (57.47 %)

Cuando ***cost*** es igual a 2.5:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.595	0.644	0.768	0.557	0.447

2_drugs_smoothed/normalized.arff **|** Neural networks (functions/MultilayerPerceptron)

learningRate	Correct	Incorrect

0.4	212 (99.06 %)	2 (0.93 %)
0.3	210 (98.13 %)	4 (1.86 %)
0.2	212 (99.06 %)	2 (0.93 %)

Cuando ***learningRate*** es igual a 0.3:

	drugA	drugB	drugC	drugX	drugY
ROC area	1.0	1.0	1.0	1.0	1.0

2_drugs_smoothed/normalized.arff **|** Método basado en ejemplos (lazy/IBK)

distanceFunction	Correct	Incorrect
Chebyshev	182 (85.04 %)	32 (14.95 %)
Euclidean	187 (87.38 %)	27 (12.61 %)
Filtered	127 (59.34 %)	87 (40.65 %)
Manhattan	188 (87.85 %)	26 (12.14 %)
Minkowski	187 (87.38 %)	27 (12.61 %)

Cuando ***distanceFunction*** es Manhattan:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.894	0.896	0.988	0.912	0.882

2_drugs_smoothed/normalized.arff **|** Regression logística (functions/SimpleLogistic)

Correct	Incorrect
209 (97.66 %)	5 (2.33 %)

	drugA	drugB	drugC	drugX	drugY
ROC area	1.0	0.997	1.0	1.0	0.998

3_drugs_discretized.arff **|** Árbol de decisión (trees/J48)

ConfidenceFactor	minNumObj	Correct	Incorrect
0.4	1	187 (87.38 %)	27 (12.61 %)
0.25	2	186 (86.91 %)	28 (13.08 %)
0.1	1	184 (85.98 %)	30 (14.01 %)
0.05	1	174 (81.30 %)	40 (18.67 %)
0.01	1	157 (73.36 %)	57 (26.63 %)

Cuando ***confidenceFactor*** es igual a 0.4:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.953	0.868	0.979	0.980	0.907

3_drugs_discretized.arff **|** Métodos bayesianos (bayes/NaiveBayes)

Correct	Incorrect
184 (85.98 %)	30 (14.01 %)

	drugA	drugB	drugC	drugX	drugY
ROC area	0.978	0.989	0.996	0.984	0.957

3_drugs_discretized.arff **|** support vector machine (functions/SMO)

Complexity	Correct	Incorrect
60.0	190 (88.78 %)	24 (11.21 %)
30.0	190 (88.78 %)	24 (11.21 %)
2.5	190 (88.78 %)	24 (11.21 %)
2.0	191 (89.25 %)	23 (87.25 %)
1.5	190 (88.78 %)	24 (11.21 %)
1.0	189 (88.31 %)	25 (11.68 %)
0.5	181 (84.57 %)	33 (15.42 %)
0.0	91 (42.52 %)	123 (57.47%)

Cuando ***complexity*** es igual a 60.0:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.974	0.969	0.985	0.961	0.947

3_drugs_smoothed/normalized.arff **|** support vector machine (functions/libSVM)

Cost	Correct	Incorrect
2.5	176 (82.24 %)	38 (17.75 %)
2.0	175 (81.77 %)	39 (18.22 %)
1.5	175 (81.77 %)	39 (18.22 %)

1.0	169 (78.97 %)	45 (21.02 %)
0.5	154 (71.96 %)	60 (28.03 %)

Cuando ***cost*** es igual a 2.5:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.760	0.651	0.961	0.887	0.870

3_drugs_discretized.arff | Neural networks (functions/MultilayerPerceptron)

learningRate	Correct	Incorrect
0.4	201 (93.92 %)	13 (6.07 %)
0.3	199 (92.99 %)	15 (7.00 %)
0.2	201 (92.92 %)	13 (6.07 %)

Cuando ***learningRate*** es igual a 0.4:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.994	0.992	0.999	0.985	0.982

3_drugs_discretized.arff | Método basado en ejemplos (lazy/IBK)

distanceFunction	Correct	Incorrect
Chebyshev	124 (57.94 %)	90 (42.05 %)
Euclidean	146 (68.22 %)	68 (31.77 %)
Filtered	139 (64.95 %)	75 (35.04 %)
Manhattan	145 (67.75 %)	69 (32.24 %)
Minkowski	146 (68.22 %)	68 (31.77 %)

Cuando ***distanceFunction*** es Manhattan:

	drugA	drugB	drugC	drugX	drugY
ROC area	0.799	0.677	0.954	0.817	0.783

3_drugs_discretized.arff | Regression logística (functions/SimpleLogistic)

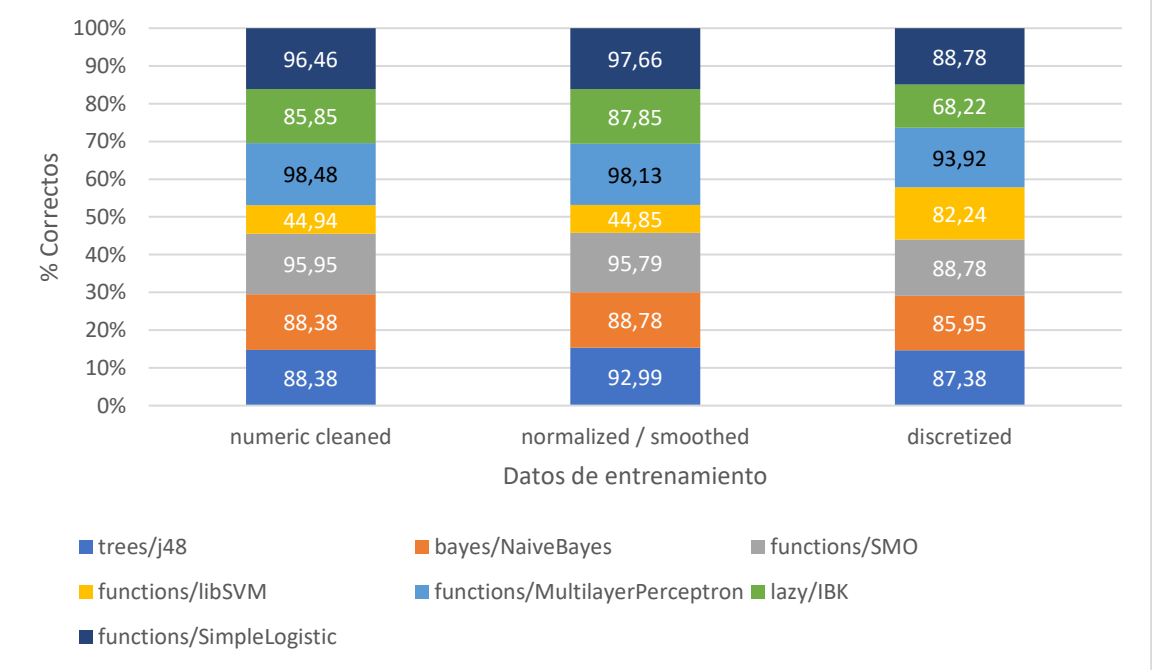
Correct	Incorrect
190 (88.78 %)	24 (11.21 %)

	drugA	drugB	drugC	drugX	drugY
ROC area	0.995	0.991	0.998	0.989	0.975

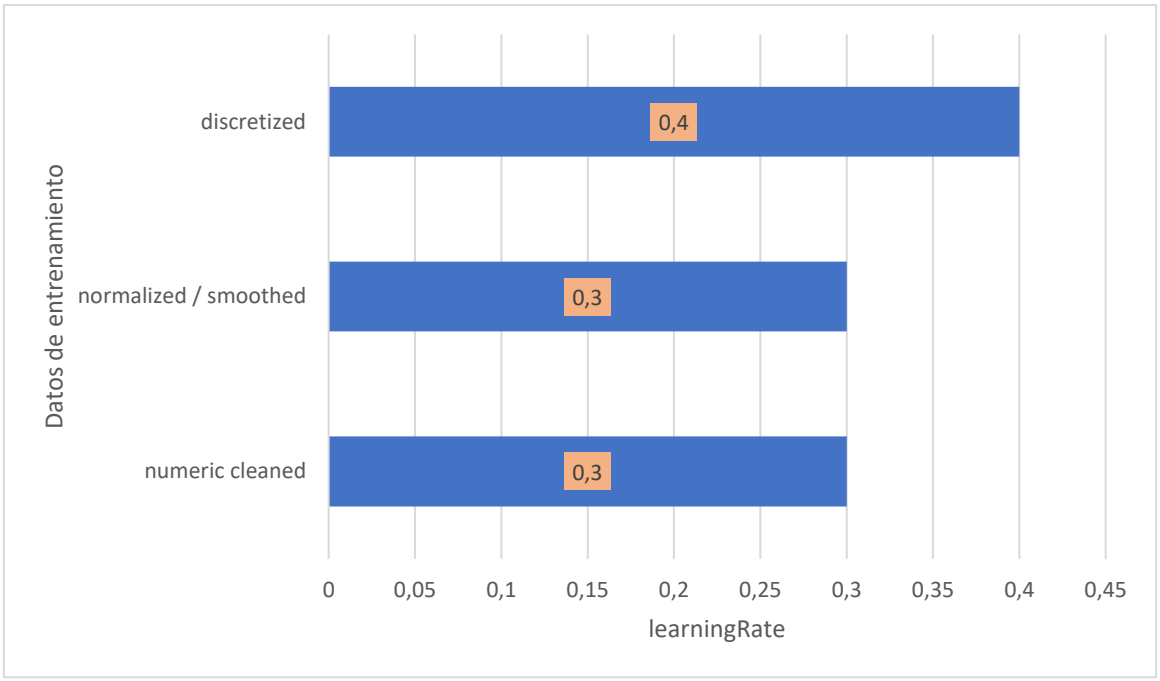
y decían que sería aburrido... estadística aplicada jajaj

JUSTIFICACION

Los 6 métodos aplicados se seleccionaron teniendo en cuenta su capacidad de resolver problemas de clasificación y se entrenaron en conjuntos de datos preparados de 4 formas diferentes para probar la efectividad de cada método en distintas configuraciones de los datos de entrenamiento. A continuación, se enlista un resumen de los resultados obtenidos con cada conjunto de datos.



De esta forma se puede evidenciar que, dado los datos originales, la mejor preparación consiste en primeramente eliminar los datos atípicos, para luego suavizarlos, así mismo se pudo concluir que discretizar los datos disminuyó la efectividad de todos los modelos excepto la de libSVM, el cual presento una mejora considerable cuando se discretizaron los datos. El mejor modelo fue producido por la red neuronal en el conjunto preparado de entrenamiento "*numeric cleaned*". MultilayerPerceptron produjo los mejores resultados en cada conjunto de datos. Se expondrá a continuación las distintas configuraciones del MultilayerPerceptron que produjeron los resultados mostrados arriba.


Datos de entrenamiento
discretized
normalized / smoothed
numeric cleaned
0,4
0,3
0,3
0
0,05
0,1
0,15
0,2
0,25
0,3
0,35
0,4
0,45
learningRate