

Thyroid Disease Data Set Analysis

Ulises Jeremias Cornejo Fandos¹ and Gaston Gustavo Rios²

¹*Licenciatura en Informatica - 13566/7, Facultad de Informatica, UNLP*

²*Licenciatura en Informatica - 13591/9, Facultad de Informatica, UNLP*

compiled: August 4, 2018

Resumen

OCIS codes:

<http://dx.doi.org/10.1364/XX.99.099999>

1. Introducción

1.A. Thyroid Disease Data Set

El data set cuenta con un conjunto de 6 bases de datos. En general, estos conjuntos son muy similares y presentan muchos atributos, *aproximadamente 29 atributos cada conjunto de datos*, siendo la mayoría booleanos o reales. El dominio del problema es el análisis de enfermedades de tiroides y se utiliza para su estudio un conjunto de datos otorgado por Garavan Institute. Cada conjunto de datos cuenta con aproximadamente 2800 ejemplos y una *gran cantidad de datos faltante*.

El conjunto de datos seleccionado permite la clasificación de pacientes entre aquellos que tienen hipotiroidismo y aquellos que no. En su versión presenta un total de 3163 ejemplares de los cuales se conocen 26 atributos.

Información de los atributos

- *age*: Este atributo corresponde a la edad del paciente. En un valor numérico continuo que toma valores de R .
- *sex*: Este atributo corresponde al sexo del paciente. Toma valores del conjunto $\{M, F\}$ y presenta valores faltante.
- *on_thyroxine*: *Thyroxine* es la principal hormona generada por la glándula tiroides. Cuando se detecta mucha thyroxine y triyodotironina entonces reduce la producción de hormonas (*TSH*). Relacionado con goitre, este es una forma de reconocer demasiada thyroxine, la cual es causada por hipertiroidismo. Muy poca es causada por hipotiroidismo. Toma valores del conjunto $\{f, t\}$.
- *query_on_thyroxine*: Este atributo indica si fue medido *on_thyroxine*. Toma valores del conjunto $\{f, t\}$.
- *on_antithyroid_medication*: Antithyroid es un medicamento que actúa sobre las hormonas tiroides. Este atributo indica si el paciente toma dicha medicación. Toma valores del conjunto $\{f, t\}$.
- *thyroid_surgery*: Este atributo indica si el paciente ha tenido una cirugía de tiroides. Toma valores del conjunto $\{f, t\}$.
- *query_hypothyroid*: Este atributo indica si le fue consultado al paciente si tenía hipotiroidismo. Toma valores del conjunto $\{f, t\}$.
- *query_hyperthyroid*: Este atributo indica si le fue consultado al paciente si tenía hipertiroidismo. Toma valores del conjunto $\{f, t\}$.
- *pregnant*: Este atributo indica si el paciente esta transitando un embarazo. Toma valores del conjunto $\{f, t\}$.
- *sick*: Este atributo indica si el paciente estaba enfermo al tomar los datos. Toma valores del conjunto $\{f, t\}$.
- *tumor*: Este atributo indica si el paciente tenía un tumor al tomar los datos. Toma valores del conjunto $\{f, t\}$.
- *lithium*: Toma valores del conjunto $\{f, t\}$.
- *goitre*: Toma valores del conjunto $\{f, t\}$.
- *TSH_measured*: Este atributo indica si se mide el *TSH* del paciente. Toma valores del conjunto $\{f, t\}$.
- *TSH*: Toma valores de R .
- *T3_measured*: Este atributo indica si se mide el *T3* del paciente. Toma valores del conjunto $\{f, t\}$.
- *T3*: Toma valores de R .

- *TT4_measured*: Este atributo indica si se mide el *TT4* del paciente. Toma valores del conjunto $\{f, t\}$.
- *TT4*: Toma valores de R .
- *T4U_measured*: Este atributo indica si se mide el *T4U* del paciente. Toma valores del conjunto $\{f, t\}$.
- *T4U*: Toma valores de R .
- *FTI_measured*: Este atributo indica si se mide el *FTI* del paciente. Toma valores del conjunto $\{f, t\}$.
- *FTI*: Toma valores de R .
- *TBG_measured*: Este atributo indica si se mide el *TBG* del paciente. Toma valores del conjunto $\{f, t\}$.
- *TBG*: Toma valores de R .

1.B. Recolección del conjunto de datos

El conjunto de datos seleccionado se obtiene en el siguiente repositorio. Como se menciona anteriormente, en el mismo existe una gran cantidad de bases de datos y del mismo se opta por utilizar un conjunto de datos agregado más recientemente.

En la descripción del mismo se duda de la integridad de los datos pero se comenta que de igual forma el otorgante es Garavan Institute al igual que el resto de los data sets. A su vez, no dice en ningún momento como es que estos datos son recolectados por lo que al momento de analizar estos datos, solo conocemos las fuentes utilizadas y la entidad otorgante.

2. Pre-procesamiento de datos

2.A. Conjunto a analizar

Como se explica en la sección anterior, existen atributos en los cuales se muestra si fue medido un atributo o no. Dado que los valores de estos coinciden con los datos faltantes se opta por la eliminación de dichos atributos. Al mismo tiempo, la cantidad de datos faltantes para la columna correspondiente al atributo **TBG** es **PORCENTAJE y SOLUCION**. Teniendo en cuenta esto, se opta por la eliminación de dicha columna dado que se considera que la misma podría llegar a interferir en el proceso del análisis de los datos, quedando finalmente un total de 19 atributos y 3163 ejemplares.

Luego se resuelven los datos faltantes de cada atributo utilizando la media de los mismos para los datos continuos y el valor de mayor frecuencia para los valores nominales. Se mapea el valor correspondiente al atributo sexo para que tome valores del conjunto $\{Male, Female\}$, y aquellos atributos booleanos para

que los valores pasen de f a *False* y de t a *True*.

3. Marco Teórico

En esta sección se introduce brevemente conceptos básicos necesarios para abordar los contenidos de las siguientes secciones del informe con mayores referencias y capacidad de entendimiento.

3.A. Árbol de decisión

Modelo descriptivo que, por su forma jerárquica, permite visualizar la organización de los atributos. Se produce a partir de la identificación sucesiva de atributos relevantes. El atributo correspondiente a la clase es cualitativo.

3.B. Árbol de clasificación

Es un Árbol de decisión cuyas hojas se refieren al mismo atributo y es discreto.

3.C. Reglas de clasificación

Modelo supervisado que a partir de la información busca obtener reglas de la forma: $A \rightarrow B$.

Comparado con el modelo explicado en la subsección anterior, las reglas de clasificación son más compactas que los árboles en donde cada regla puede representar un concepto distinto permitiendo así agregar o quitar reglas dinámicamente.

3.D. Clustering

Un algoritmo de agrupamiento (en inglés, clustering) es un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio. Esos criterios son por lo general distancia o similitud. La cercanía se define en términos de una determinada función de distancia, como la euclídea, aunque existen otras más robustas o que permiten extenderla a variables discretas. La medida más utilizada para medir la similitud entre los casos es la matriz de correlación entre los $n \times n$ casos. Sin embargo, también existen muchos algoritmos que se basan en la maximización de una propiedad estadística llamada verosimilitud.

En el contexto de la Minería de Datos se lo considera una técnica de aprendizaje no supervisado puesto que busca encontrar relaciones entre variables descriptivas pero no la que guardan con respecto a una variable objetivo.

3.D.1. K-Means

K-means es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano.

En la figura 1 se muestra un ejemplo de aplicación del algoritmo k-means para realizar un agrupamiento de los datos.



Fig. 1. Ejemplo de aplicación del algoritmo k-means para realizar un agrupamiento de los datos.

3.E. Redes Neuronales

Las **redes neuronales** son un modelo computacional basado en un gran conjunto de unidades neuronales simples, **neuronas artificiales**, de forma aproximadamente análoga al comportamiento observado en los axones de las neuronas en los cerebros biológicos.

Cada unidad neuronal está conectada con muchas otras y los enlaces entre ellas pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Cada unidad neuronal, de forma individual, opera empleando funciones de suma. Puede existir una función limitadora o umbral en cada conexión y en la propia unidad, de tal modo que la señal debe sobrepasar un límite antes de propagarse a otra neurona.

Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, y sobresalen en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional.

3.E.1. Perceptron

En el campo de las Redes Neuronales, el **perceptrón**, se refiere a la neurona artificial o unidad básica de inferencia en forma de discriminador lineal, a partir del cual se desarrolla un algoritmo capaz de generar un criterio para seleccionar un sub-grupo a partir de un grupo de componentes más grande.

En las figuras 2 y 3 se muestra la arquitectura que determina el comportamiento de un perceptron.

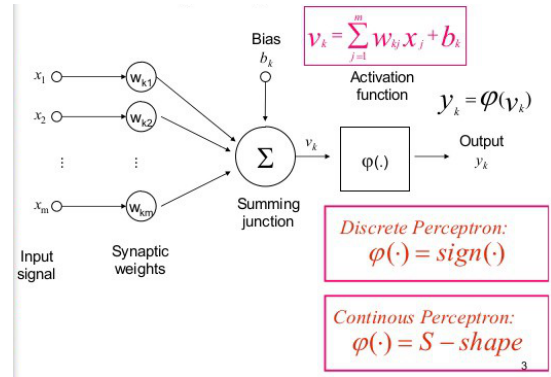


Fig. 2. Arquitectura de un perceptron.

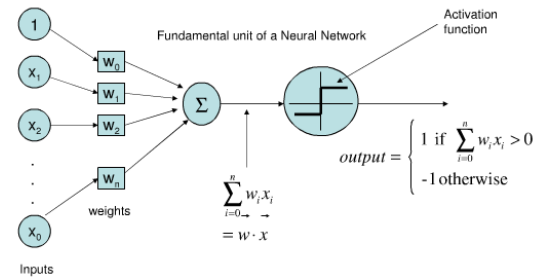


Fig. 3. Arquitectura de un perceptron.

La limitación de este algoritmo es que si dibujamos en un gráfico estos elementos, se deben poder separar con un hiperplano únicamente los elementos "deseados" discriminándolos (ó *separándolos*) de los "no deseados" como se muestra en la figura 4.

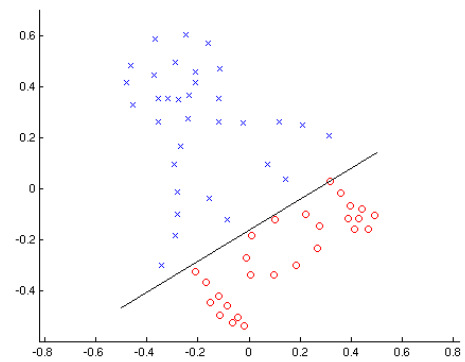


Fig. 4. Ejemplo de función lineal discriminante.

El perceptrón puede utilizarse con otros tipos de perceptrones o de neurona artificial, para formar una red neuronal artificial más compleja.

4. Análisis de datos

Para el análisis de los datos se evalúan distintas gráficas de lo mismos, como histogramas de los datos nominales y gráficas de dispersión para aquellos datos de tipo numérico, además de ciertas métricas que permiten conocer la correlación entre cada uno de ellos. De este modo se permite observar relaciones entre los distintos atributos del conjunto de datos así como también la relación entre estos mismos y la etiqueta, o *label*.

Se dispone de las gráficas correspondientes a los atributos en la sección 10.A.1 del apéndice.

Posteriormente, se calcula el índice de correlación lineal entre los atributos, para comenzar así con el análisis de las relaciones entre cada uno de ellos. El cálculo de los mismos se ve reflejado en la siguiente tabla (1).

Atributos	age	TSH	T3	TT4	TU4	FTI
age	1	-0.007	-0.269	-0.091	-0.194	0.015
TSH	-0.007	1	-0.172	-0.310	0.069	-0.244
T3	-0.269	-0.172	1	0.545	0.388	0.294
TT4	-0.091	-0.310	0.545	1	0.323	0.685
T4U	-0.194	0.069	0.388	0.323	1	-0.284
FTI	0.015	-0.244	0.294	0.685	-0.283	1

Table 1. Matriz de correlación lineal

Como se observa en la tabla 1, las tuplas (FTI, TT4) y (T3, TT4) presentan una correlación lineal leve, con un índice de correlación de 0.685 y 0.545 respectivamente.

5. Método Experimental

En esta sección se detalla todo lo referido al estudio y la creación de los distintos modelos de sistemas inteligentes utilizados para el estudio del conjunto de datos elegido.

5.A. Árbol de decisión

Para la creación del modelo de Árbol de decisión, se evalúan las distintas posibilidades permitiendo así la construcción de un modelo con un mayor nivel de cobertura sobre el conjunto de datos empleado para entrenamiento y prueba.

EXPLICAR ELECCIÓN DEL MODELO

Por lo tanto, como se menciona anteriormente, dado que el conjunto de datos presenta atributos numéricos de tipo continuo, no es completamente viable discretizar los mismos en intervalos si es que existe alguna forma de construir un modelo de Árbol de decisión evitando esto.

Se opta finalmente la utilización del algoritmo C4.5 para generar el Árbol de decisión. Se genera el Árbol utilizando el operador W-J48 de rapidminer, utilizando la configuración por defecto de la mayoría de los flags exceptuando el flag C, *confianza*. Luego de probar distintas configuraciones para dicho flag, se observa que dado un conjunto de entrenamiento del 70% del total de los datos, con un 30% de los datos destinado al testing del modelo, el porcentaje de acierto del modelo es 99.59% cuando $C \geq 0.5$, por lo que se configura el flag con C con el valor 0.5.

En la figura (5) se puede observar el modelo obtenido utilizando el algoritmo C4.5.

W-J48

J48 pruned tree

```

FTI <= 63
| TSH <= 6.2: negative (41.0/2.0)
| TSH > 6.2
| | on_thyroxine = False: hypothyroid (90.0/5.0)
| | on_thyroxine = True
| | | TSH <= 23: negative (3.0)
| | | TSH > 23: hypothyroid (12.0/1.0)
FTI > 63
| TSH <= 6.3: negative (1917.0/1.0)
| TSH > 6.3
| | age <= 59: negative (94.0/2.0)
| | age > 59
| | | FTI <= 69
| | | | T4U <= 0.97: hypothyroid (3.0)
| | | | T4U > 0.97: negative (4.0)
| | | | FTI > 69: negative (50.0/2.0)

```

Number of Leaves : 9

Size of the tree : 17

Fig. 5. Modelo de Árbol generado utilizando el algoritmo C4.5, con una performance de 99.16%.

En la figura (29) de la sección 10.B.2 se puede observar la performance del modelo obtenido aplicando el mismo sobre un conjunto de testing.

5.B. Reglas de Clasificación

Para definir el algoritmo a utilizar para la creación de reglas se evalúa cada uno de ellos comparando los modelos generados para determinar así cual es más conveniente. Entre los algoritmos evaluados están OneR y PRISM.

5.B.1. OneR

Para la construcción de este modelo se trabaja con un conjunto normalizado de los datos utilizando normalización Z sobre los datos de tipo numérico. El antecedente de la regla generada por este algoritmo se define a partir del atributo *FTI* como se puede observar

en la figura (6).

W-OneR

```
FTI:
    < -1.1047537479703995 -> hypothyroid
    < -0.94914920044515   -> negative
    < -0.9232151091909417 -> hypothyroid
    >= -0.9232151091909417 -> negative
(2169/2214 instances correct)
```

Fig. 6. Modelo generado por el algoritmo OneR.

En la figura (30) de la sección 10.B.2 se puede observar la performance del modelo obtenido aplicando el mismo sobre un conjunto de testing.

5.B.2. PRISM

Para la construcción del modelo de reglas utilizando el algoritmo PRISM se discretiza por frecuencia los datos numéricos probando le performance del modelo para distintos intervalos. Finalmente se opta por discretizar en 7 intervalos obteniendo los resultados que se muestran en la figura (31) de la sección 10.B.2.

6. Clustering

6.A. K-Means

Para la construcción de este modelo se utiliza el algoritmo K-means evaluando las distancias con distancia euclídea. Se evalúa el modelo resultante utilizando el índice de Davies Bouldin notando que se obtiene mejores resultados cuando el número de clusters $k = 2$. Finalmente el índice de Davies Bouldin obtenido es 0.601.

En el agrupamiento resultante se pueden ver dos agrupamientos, *cluster_0* y *cluster_1* en los que se ve que si el paciente pertenece al *cluster_0* no tiene hipotiroides. Al mismo tiempo, las instancias pertenecientes al *cluster_0* tienen TSH mas bajo, T3 mas alto, TT4 bastante mas alto y FTI bastante mas alto. Hay muy pocos que son de *cluster_0* y se pueden ver los datos del agrupamiento en la tabla (2).

hypothyroid	cluster	cantidad
hypothyroid	cluster_1	151.0
negative	cluster_0	39.0
negative	cluster_1	2973.0

Table 2. Resultados del clustering generado con k.means

Los datos referidos al agrupamiento se pueden ver en la sección 10.B.2 del apéndice.

7. Redes Neuronales

Se evalúan distintos modelos de redes neuronales con distintos conjuntos de prueba para evaluar así la performance de los mismos y comparar los modelos resultantes.

7.A. Perceptrón

Dado que el atributo etiqueta del data set es un es un conjunto de valores categóricos de cardinalidad igual a 2, se evalúa la utilización de un perceptrón como posible arquitectura de redes neuronales.

Pesos y bias

Como se menciona en el marco teórico, la función discriminante que define al perceptron se define en función de un Intercept y distintos pesos o coeficientes asociados a cada uno de los valores de los atributos del data set. A continuación se enlistan los valores obtenidos luego de entrenar el perceptron.

aplicando el mismo sobre un conjunto de testing.

$$Intercept = 1.3490659272630658$$

(1)

$$w(sex = Male) = -0.006$$

(2)

$$w(sex = Female) = 0.006$$

(3)

$$w(on_thyroxine = False) = -0.002$$

(4)

$$w(on_thyroxine = True) = 0.002$$

(5)

$$w(query_on_thyroxine = False) = -0.203$$

(6)

$$w(query_on_thyroxine = True) = 0.203$$

(7)

$$w(on_antithyroid_medication = False) = -0.018$$

(8)

$$w(on_antithyroid_medication = True) = 0.018$$

(9)

$$w(thyroid_surgery = False) = 0.009$$

(10)

$$w(thyroid_surgery = True) = -0.009$$

(11)

$$w(query_hypothyroid = False) = -0.046$$

(12)

$$w(query_hypothyroid = True) = 0.046$$

(13)

$$w(query_hyperthyroid = False) = -0.002$$

(14)

$$w(query_hyperthyroid = True) = 0.002$$

(15)

$$w(pregnant = False) = -0.052$$

(16)

$$w(pregnant = True) = 0.052$$

(17)

$$w(sick = False) = -0.019$$

(18)

$$w(sick = True) = 0.019$$

(19)

$$w(tumor = False) = -0.295$$

(20)

$$w(tumor = True) = 0.295$$

(21)

$$w(lithium = False) = -1.955$$

(22)

$$w(lithium = True) = 1.955$$

(23)

$$w(goitre = False) = 0.001$$

(24)

$$w(goitre = True) = -0.001$$

(25)

$$w(age) = -0.016$$

(26)

$$w(TSH) = 0.007$$

(27)

$$w(T3) = -0.025$$

(28)

$$w(TT4) = 0.232$$

(29)

$$w(T4U) = -0.102$$

(30)

$$w(FTI) = 0.438$$

(31)

7.B. Multiperceptron

Se entrena un multiperceptron utilizando el operador NeuralNet de RapidMiner con 500 ciclos de entrenamiento, 0.03 de tasa de aprendizaje y momentum igual a 0.2, obteniendo una performance mayor a la obtenida con el modelo anterior.

En las figuras (34) y (35) de la seccion 10.B.2 se puede observar la performance del modelo de redes neuronales obtenido aplicando el mismo sobre un conjunto de testing.

8. Análisis de Resultados

ANALISIS DE RESULTADOS

9. Discusión y Conclusiones

DISCUSION Y CONCLUSIONES

References

- [1] <https://authors.aps.org/revtex4/>.
- [2] http://www.opticsinfobase.org/submit/style/jrnls_style.cfm.

En las figuras (32) y (33) de la seccion 10.B.2 se puede observar la performance del modelo obtenido

10. Apéndice

10.A. Imágenes

10.A.1. Gráficos de los atributos

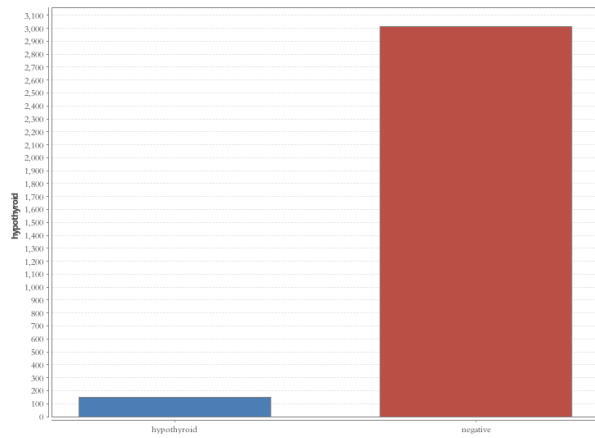


Fig. 7. Gráfico de barras del atributo etiqueta.

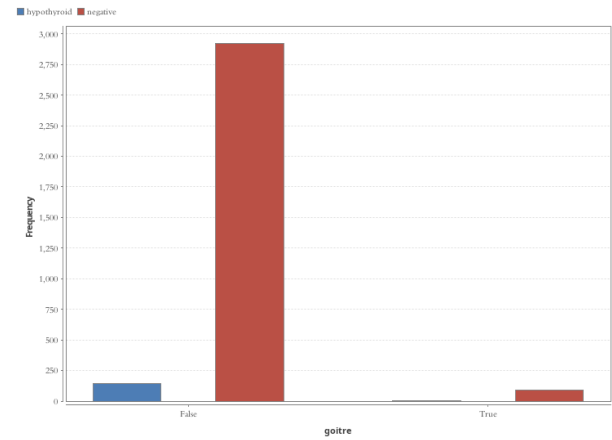


Fig. 10. Histograma del atributo *goitre*.

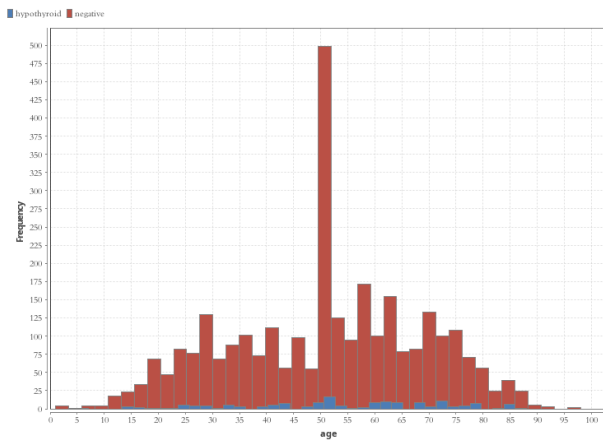


Fig. 8. Histograma del atributo *age*.

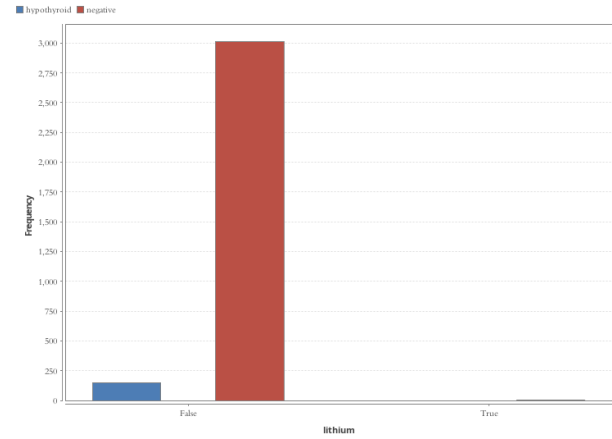


Fig. 11. Histograma del atributo *lithium*.

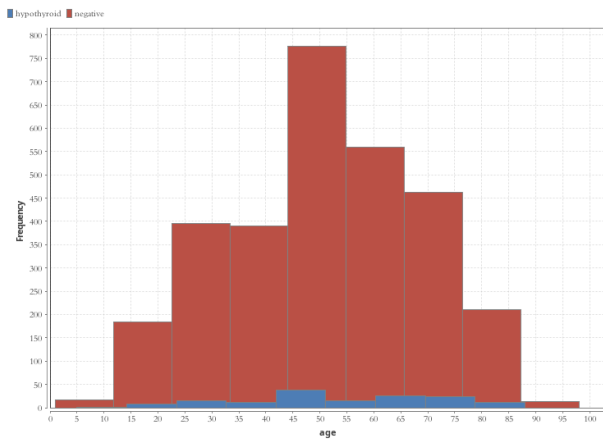


Fig. 9. Histograma del atributo *age*.

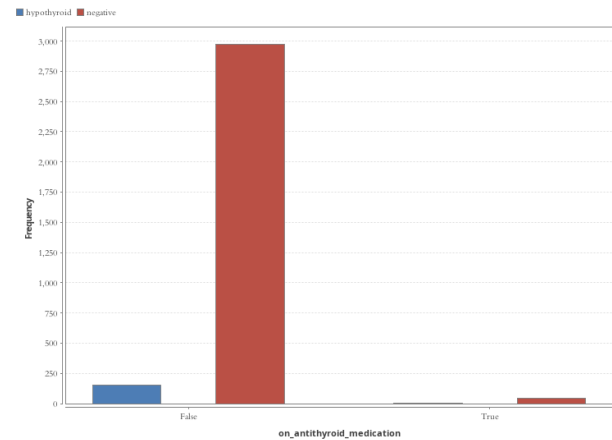
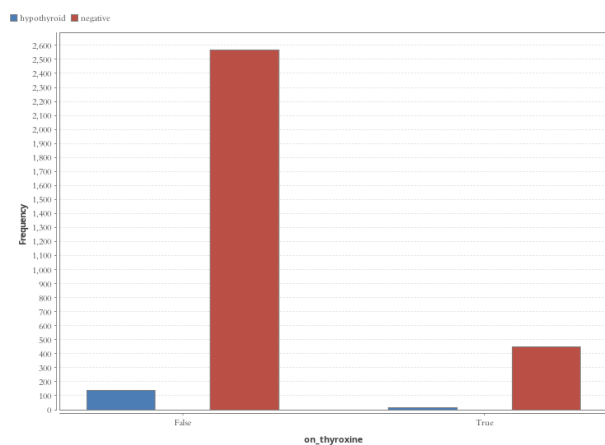
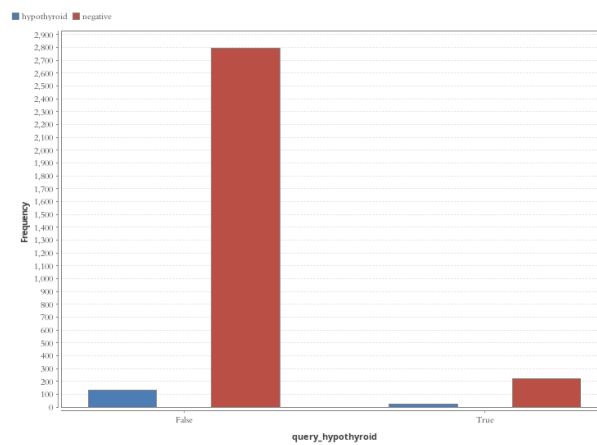
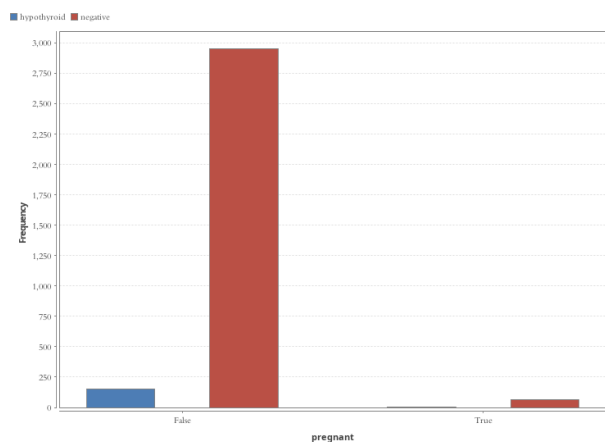
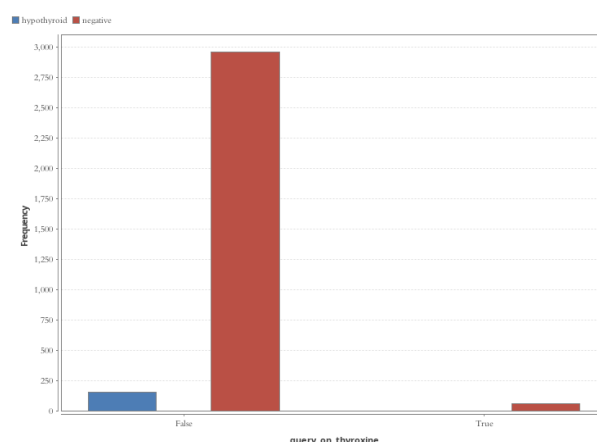
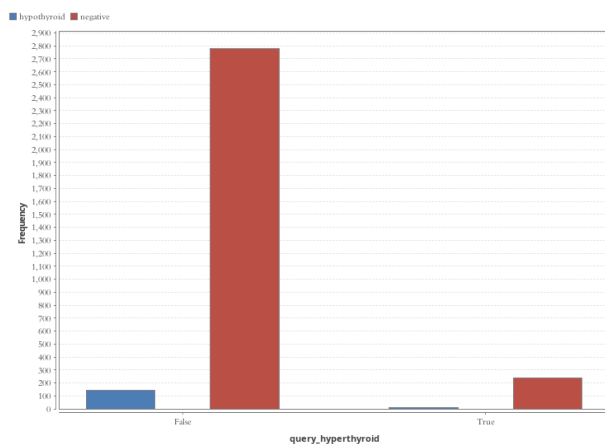
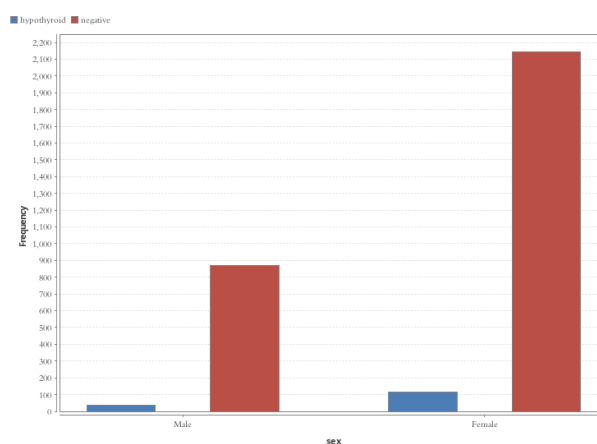


Fig. 12. Histograma del atributo *on_antithyroid_medication*.

Fig. 13. Histograma del atributo *on_thyroxine*.Fig. 16. Histograma del atributo *query_hypothyroid*.Fig. 14. Histograma del atributo *pregnant*.Fig. 17. Histograma del atributo *query_on_thyroxine*.Fig. 15. Histograma del atributo *query_hyperthyroid*.Fig. 18. Histograma del atributo *sex*.

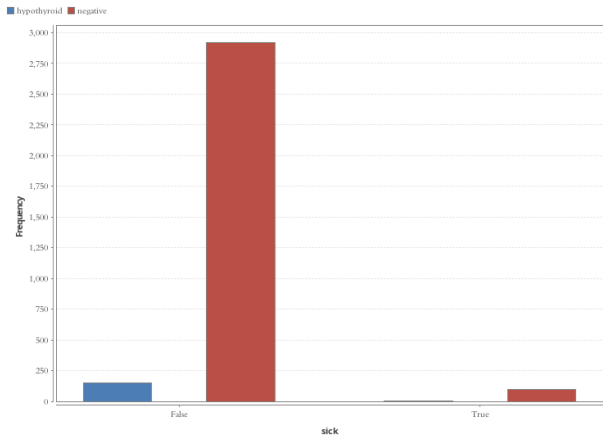


Fig. 19. Histograma del atributo *sick*.

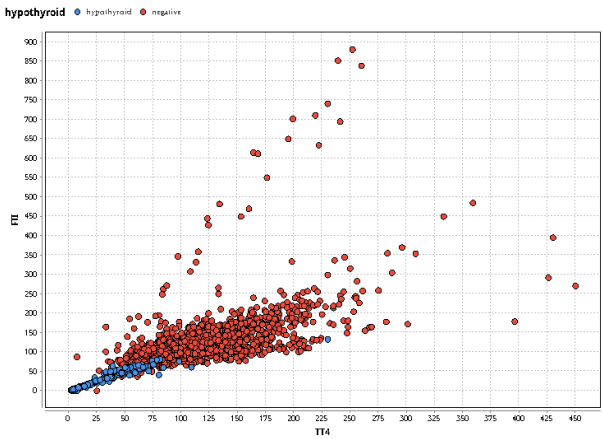


Fig. 22. Diagrama de dispersión de los atributos *TT4*, *FTI*.

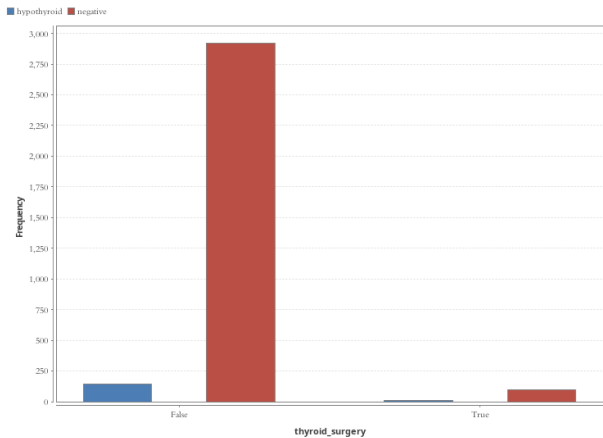


Fig. 20. Histograma del atributo *thyroid_surgery*.

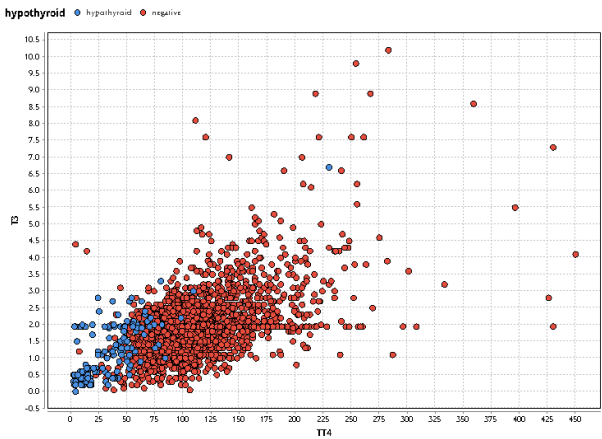


Fig. 23. Diagrama de dispersión de los atributos *TT4*, *T3*.

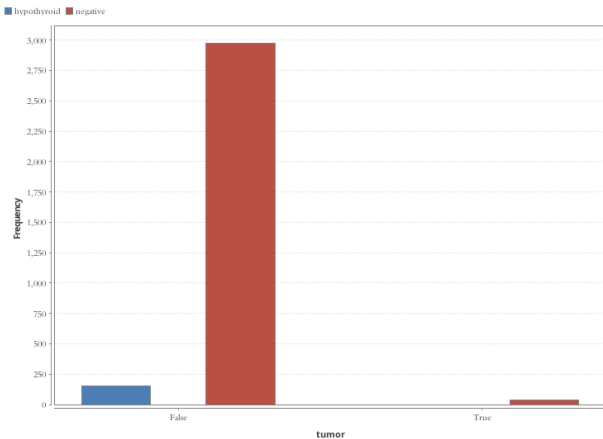


Fig. 21. Histograma del atributo *tumor*.

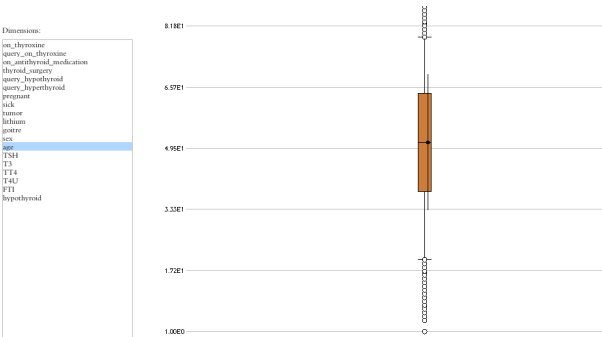
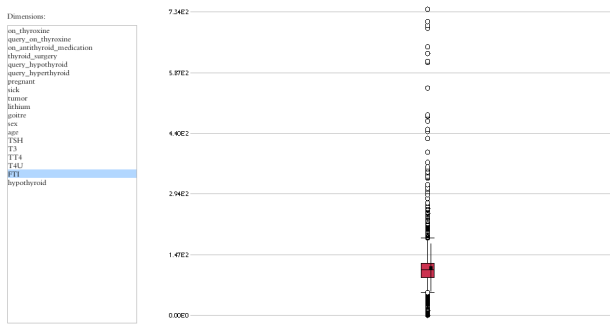
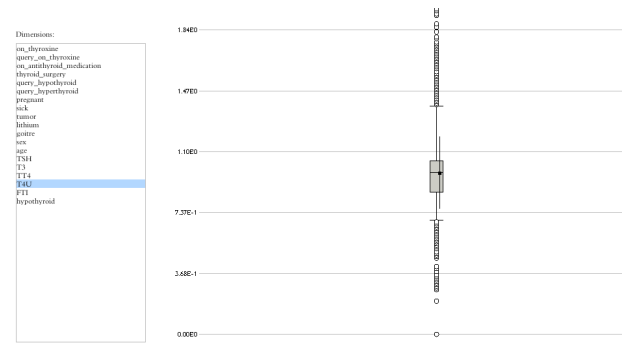
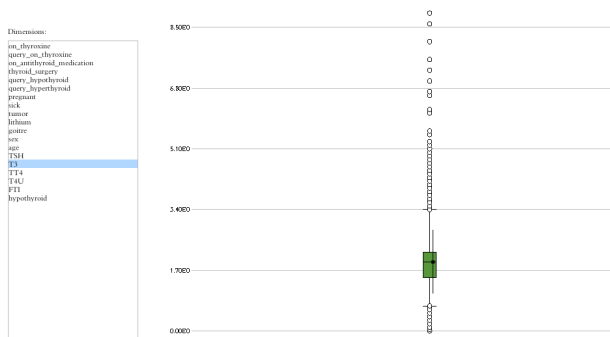
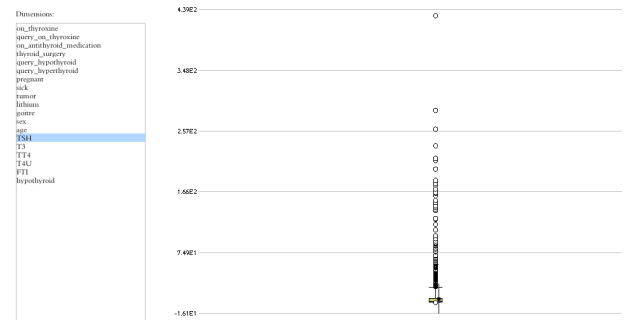


Fig. 24. Diagrama de cajas del atributo *age*.

Fig. 25. Diagrama de cajas del atributo *FTI*.Fig. 27. Diagrama de cajas del atributo *T4U*.Fig. 26. Diagrama de cajas del atributo *T3*.Fig. 28. Diagrama de cajas del atributo *TSH*.

10.B. Modelos y Performance

10.B.1. Modelos

Atributo	cluster_0	cluster_1
sex = M	0.10256410256410256	0.2893725992317542
sex = F	0.8974358974358975	0.7106274007682458
on_thyroxine = f	0.9487179487179487	0.8530729833546735
on_thyroxine = t	0.05128205128205128	0.1469270166453265
query_on_thyroxine = f	1.0	0.9823943661971831
query_on_thyroxine = t	0.0	0.017605633802816902
on_antithyroid_medication = f	1.0	0.9865556978233034
on_antithyroid_medication = t	0.0	0.013444302176696543
thyroid_surgery = f	1.0	0.9667093469910372
thyroid_surgery = t	0.0	0.03329065300896287
query_hypothyroid = f	0.9487179487179487	0.9234955185659411
query_hypothyroid = t	0.05128205128205128	0.0765044814340589
query_hyperthyroid = f	0.6923076923076923	0.926056338028169
query_hyperthyroid = t	0.3076923076923077	0.07394366197183098
pregnant = f	0.9743589743589743	0.9801536491677336
pregnant = t	0.02564102564102564	0.019846350832266324
sick = f	1.0	0.9683098591549296
sick = t	0.0	0.03169014084507042
tumor = f	1.0	0.9871959026888605
tumor = t	0.0	0.012804097311139564
lithium = f	1.0	0.9993597951344431
lithium = t	0.0	6.402048655569782E-4
goitre = f	1.0	0.9683098591549296
goitre = t	0.0	0.03169014084507042
age	51.51282051282051	51.127720870678615
TSH	4.964323581180723	5.935150321052643
T3	3.487115072933549	1.920431471620351
TT4	249.64102564102564	107.09236555697822
T4U	0.616923076923077	0.9827091372498201
FTI	472.56410256410254	110.93890826408756

Table 3. Agrupamiento generado por el algoritmo *k-means*

10.B.2. Performance

accuracy: 99.16%

	true hypothyroid	true negative	class precision
pred. hypothyroid	40	3	93.02%
pred. negative	5	901	99.45%
class recall	88.89%	99.67%	

Fig. 29. Performance del modelo de Árbol generado utilizando el algoritmo C4.5 sobre un conjunto de testeo correspondiente al 30% del conjunto total de datos.

accuracy: 98.10%

	true hypothyroid	true negative	class precision
pred. hypothyroid	39	12	76.47%
pred. negative	6	892	99.33%
class recall	86.67%	98.67%	

Fig. 30. Performance del modelo de reglas generado utilizando el algoritmo OneR sobre un conjunto de testeo correspondiente al 30% del conjunto total de datos.

accuracy: 97.26%

	true hypothyroid	true negative	class precision
pred. hypothyroid	39	20	66.10%
pred. negative	6	884	99.33%
class recall	86.67%	97.79%	

Fig. 31. Performance del modelo de reglas generado utilizando el algoritmo PRISM sobre un conjunto de testeo correspondiente al 30% del conjunto total de datos.

accuracy: 95.79%

	true hypothyroid	true negative	class precision
pred. hypothyroid	6	1	85.71%
pred. negative	39	903	95.86%
class recall	13.33%	99.89%	

Fig. 32. Performance del modelo de redes neuronales generado utilizando la arquitectura perceptron sobre un conjunto de testeo correspondiente al 30% del conjunto total de datos.

accuracy: 97.15%

	true hypothyroid	true negative	class precision
pred. hypothyroid	65	4	94.20%
pred. negative	86	3008	97.22%
class recall	43.05%	99.87%	

Fig. 33. Performance del modelo de redes neuronales generado utilizando la arquitectura perceptron sobre un conjunto de testeo correspondiente conjunto total de datos de entrenamiento.

accuracy: 98.52%

	true hypothyroid	true negative	class precision
pred. hypothyroid	36	5	87.80%
pred. negative	9	899	99.01%
class recall	80.00%	99.45%	

Fig. 34. Performance del modelo de redes neuronales generado utilizando la arquitectura multiperceptron sobre un conjunto de testeo correspondiente al 30% del conjunto total de datos.

accuracy: 98.45%

	true hypothyroid	true negative	class precision
pred. hypothyroid	120	18	86.96%
pred. negative	31	2994	98.98%
class recall	79.47%	99.40%	

Fig. 35. Performance del modelo de redes neuronales generado utilizando la arquitectura multiperceptron sobre un conjunto de testeo correspondiente al conjunto total de datos de entrenamiento.