Minería de datos

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INTELIGENCIA ARTIFICIAL

Práctica 2: Evaluación de la relevancia de variables de entrada

Jon Etxeberria San Millán

jecheverr33@alumno.uned.es

Curso: 2021-2022

Práctica 02

Contenido

1. Introducción

En Minería de Datos, la selección de características y la evaluación de la relevancia de variables de entrada es una parte fundamental para obtener un dataset bien entrenado. La selección de características hace referencia al proceso de reducir las entradas para su procesamiento y análisis, o de encontrar las entradas más significativas. Cuando hay una gran cantidad de variables redundantes en el conjunto de datos, no solo dañará el rendimiento del modelo, sino que también aumentará el costo del modelado, por lo que la selección de características sigue siendo necesaria. La selección de variables en función de la relevancia de las mismas, ofrece al menos tres beneficios:

1. ***Reducir la posibilidad de sobreajuste:*** Con menos datos redundantes, hay menos posibilidades de tomar decisiones basadas en datos ruidosos.
2. ***Mejorar la precisión:*** Hay menos datos incorrectos y, por supuesto, los datos buenos se ajustan bien al modelo.
3. ***Reducir el tiempo de entrenamiento del modelo:*** La cantidad de datos es menor, la computadora come menos y la ejecución es feliz.

Este trabajo se basa en las referencias propuestas en la práctica, uno para la definición de la relevancia [1] y otro en las diferentes técnicas para estimar la relevancia [2]. Además se ha estudiado la el estado del arte alrededor de estas técnicas, estudiando textos de técnicas de estimación de relevancia y de técnicas de ránking y filtrado([3], [4], [5], [6], [7]).

2. Descripción de cómo se han definido las 6 variables del conjunto de datos sintético

Siguiendo el enunciado de la práctica, se han creado 5 variables de entrada y una variable de salida. Así según el enunciado, las 5 variables de entrada se crean de la siguiente forma:

1. Definir 3 variables (X_1, X_2, X_3) de forma aleatoria.
2. Extender el ejemplo de XOR a 3 dimensiones [1] a la variable salida ($Y= X_1 \oplus X_2 \oplus X_3$).
3. Definir la cuarta variable como una función de un subconjunto de las 3 primeras ($X_4= X_2$ XOR X_3).
4. La quinta como variable booleana.

De este modo, y siguiendo estas directrices, se diseña el dataset de trabajo para esta práctica.

Tabla 1: Variables del dataset de trabajo

VARIABLES DEL DATASET
VARIABLES DE ENTRADA
$\boldsymbol{X_1, X_2, X_3}$ se generan de forma aleatoria siguiendo una distribución normal
$\boldsymbol{X_4 = X_2 \oplus X_3}$
$\boldsymbol{X_5}$ se genera de forma aleatoria siguiendo una distribución normal
VARIABLE DE SALIDA O CLASE
$\boldsymbol{Y= X_1 \oplus X_2 \oplus X_3 \simeq Y = X_1 \oplus X_4}$

Al analizar la Tabla1, podemos obtener las siguientes conclusiones:

- La variable de salida o clase, tiene una fórmula equivalente: $Y = X_1 \oplus X_4$. Esto se debe a que $X_4 = X_2 \oplus X_3$.
- En las dos fórmulas de variable de salida o clase, siempre parece la variable de entrada X_1. Esto indica que esta variable es importante o indispensable para el dataset, mientras que el resto lo son menos. Habrá que estudiar la relevancia del resto de variables aplicando demostraciones probabilísticas para cada una.
- La variable X_5 se muestra totalmente irrelevante para este dataset, ya que no tiene ninguna relación con la variable de salida.

Con estas características de dataset, se crea un dataset primario donde se introducen todas las diferentes entradas posibles, aplicando las restricciones del enunciado y las conclusiones a las que se han llegado. Esta tabla básica consta de 16 entradas, y puede verse en la Ilustración1 obtenida de aplicar el algoritmo en NetBeans:

Tabla basica (16) datos

X1	X2	X3	X4	X5	Y
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	0	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	0	1
0	0	0	0	1	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	1	0
1	0	0	0	1	1
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	0	1	1

Ilustración 1: Listado de 16 instancias únicas del dataset

Para la aplicación de los métodos de selección de atributos en los siguientes puntos, el dataset se amplía has las 100 entradas. Esto se hace tomando de forma aleatoria las entradas de la Tabla básica y creando un nuevo dataset que cumpla con las restricciones del enunciado.

3. Descripción de las técnicas utilizadas para la selección de variables de entrada

Lo primero que tenemos que hacer una vez obtenido el dataset de entrada, es el análisis de relevancia de cada variable aplicando fórmulas de probabilidad. Para esto, se ha seguido el texto de Kohavi [1], donde se definen los tipos de relevancia de las variables. De esta definición y de su identificación en el dataset de las entradas débiles o fuertes, se puede en un paso posterior hacer una limpieza que mejore en términos de rapidez de computación el trabajo con el dataset y se eliminen datos que puedan generar ruido en la optimización de este.

Se desprende del texto que existen dos grados de relevancia: débil y fuerte. Para la definición de "*Relevancia*" debe definirse en términos de un clasificador óptimo de Bayes: el clasificador óptimo. Una característica X es ***fuerte*** a nivel de relevancia si la eliminación de X solo resultara en deterioro del rendimiento de un clasificador bayesiano óptimo. Una característica X es ***débil*** a nivel de relevancia si no es muy relevante y existe un subconjunto de características, S, tal que el rendimiento de un clasificador Bayes en S es peor que el rendimiento en S U {X}. Una característica es ***irrelevante*** si no es fuerte o débilmente relevante.

Definición 1. Relevancia Fuerte: Una variable Xi es relevante fuerte si existen un xi, un y, y un s_i para los cuales $p(Xi = x_i, S_i = s_i) > 0$ tal que $p(Y = y|X_i = x_i, S_i = s_i) = p(Y = y|S_i = s_i)$. La fuerte relevancia implica que la característica es indispensable en el sentido de que no se puede eliminar sin pérdida de precisión en la predicción.

Definición 2. Relevancia Débil: Por exclusión, si una variable es fuerte ya no es débil. Además, una variable X_i es relevante débil si existe un subconjunto de variables S'_i de S_i para los que existen un x_i, un y, y un s'_i para los cuales $p(X_i = x_i, S_i' = s'_i) > 0$ tal que $p(Y = y|X_i = x_i, S'_i = s'_i) = p(Y = y|S'_i = s'_i)$. La relevancia débil implica que la característica a veces puede contribuir a la precisión de la predicción.

Definición 3. Variable Irrelevante: Una variable se dice que es irrelevante, cuando no es ni relevante fuerte ni relevante débil.

Para el dataset que estamos trabajando, ya conocemos de antemano las relevancias de cada variable sin necesidad de aplicar las fórmulas planteadas por Kohavi. X_1 relevante fuerte; X_2, X_3, X_4 relevantes débiles, y X_5 irrelevante. Para dotar de rigor al trabajo, a continuación, se realiza una demostración aplicando las fórmulas de probabilidad bayesiana planteadas por Kohavi.

3.1. Metodología

La metodología que se ha seguido para la resolución de este apartado, es la de diseñar una serie de funciones que permitan crear el dataset necesario, y determinar la fortaleza o debilidad de cada variable. El desarrollo se ha implementado en Java, y el código se adjunta a la práctica. En la siguiente Tabla2 se muestran las funciones diseñadas.

Tabla 2: Algoritmos utilizados para cálculo de relevancia de variables

Algoritmo	Función
A_1	*Creación del dataset*
A_2	*Probabilidad Condicionada*
A_3	*Comprobar Relevancia Fuerte*
A_4	*Comprobar Relevancia Débil*

Para determinar la clase de variables que existen en el dataset, se han diseñado los algoritmos de Comprobación de relevancia, que se apoyan en la Probabilidad de Bayes o probabilidad Condicionada.

En el algoritmo A_2 se implementa la condición Bayesiana para apoyar a los algoritmos de relevancia. Formalmente, para dos sucesos A y B donde P(B) > 0, se define la probabilidad condicionada de A dado B como:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Se interpreta la probabilidad como la probabilidad de A suponiendo que B haya ocurrido.

Aplicando Kohavi[1], se calcula la probabilidad de las salidas con 2 subconjuntos de entradas diferentes a comparar, uno con la variable objetivo y otro subgrupo extrayendo la variable objetivo a determinar su fortaleza. Así , se usa A_2 para el cálculo de $p(Y = y|Xi = xi, Si = si) = p(Y = y|Si = si)$. Así, habrá que comparar cuando $p(X_i = x_i, Si = si) > 0$, tal que $p(Y = y \mid X_i = x_i, Si = si) \neq p(Y = y \mid Si = si)$. O lo que es lo mismo, existen dos instancias A y B que difieren únicamente en X_i y satisfacen YA != YB. Esta comparación realizada a través de A_3, nos permitirá determinar si una variable es ***relativamente fuerte***.

En cuanto a diagnosticar si una variable es ***relativamente débil***, el primer paso es demostrar que no sea relativamente fuerte. De este modo, se aplicará A_3, y en caso de no obtener respuesta positiva, se aplica A_4. Este algoritmo, genera una combinación de entradas posibles con las variables del conjunto S. Se generan todas las combinaciones posibles con el número de variables del dataset, en nuestro caso de dos, tres y cuatro variables. Se seleccionan las opciones donde no está la variable objetivo, A partir de este momento se aplica la definición de Kohavi, y se investiga si la variable objetivo es fuerte en algún subconjunto de la lista: cuales $p(Xi = xi, Si' = s'i) > 0$ tal que $p(Y = y|Xi = xi, S'i = s'i) = p(Y = y|S'i = s'i)$.

En caso de que la variable objetivo no sea relativamente fuerte o débil, se dice que se trata de una ***variable irrelevante***.

3.2. Demostración de relevancia de las variables de entrada

3.2.1. Variable de entrada X_1

La variable X_1 por enunciado debería de resultar ser relevante fuerte. Haciendo uso del algoritmo A_3 se puede comprobar el resultado. Aunque para la determinación de relevancia de esta variable, también se puede hacer de forma intuitiva y visual haciendo los cálculos con la Tabla Básica.

Tabla 3: Relevancia variable X_1

Valores entrada
$x_1 = 0$; $S = (x_2 = 0, x_3 = 0, x_4 = 0, x_5 = 0)$; $y = 1$
Cálculo de probabilidades
$p(X_1 = 0, S) = 1/16 > 0$ **$p(Y = 1 \mid X_1 = 0, S) = 0$** **$p(Y = 1 \mid S) = 1/2$**

De las probabilidades obtenidas, al aplicar la Definición1 tenemos que :

- $p(Xi = xi, Si = si) = p(X_1 = 0, S = (x_2 = 0, x_3 = 0, x_4 = 0, x_5 = 0) = 1/16 > 0$

- $p(Y = y|Xi = xi, Si = si) = p(Y = y|Si = si) \longrightarrow p(Y = 1|X1 = 0, S) = p(Y = 1|S) \longrightarrow 0 \neq 1/2$

Por lo tanto, ***Definición1 == True;***

3.2.2. Variable de entrada X_2

La variable X_2 por enunciado debería de resultar ser relevante débil. Haciendo uso de los algoritmos A_3 primero para descartar que se trata de una variable de relevancia fuerte y A_4 después, se puede comprobar el resultado de relevancia débil de la variable X_2. También al igual que en el caso anterior, se puede hacer una aproximación al problema de forma intuitiva y visual sobre la Tabla básica de entradas de instancias.

Tabla 4: Relevancia de variable X_2

Valores entrada
$X_2 = 0$; $S' = S - \{X_4\} \cong S' = (x_1 = 0, x_3 = 0, x_5 = 0)$; $y = 1$
Cálculo de probabilidades
$p(X_2 = 0, S') = 1/16 > 0$ **$p(Y = 1 \mid X_2 = 0, S') = 0$** **$p(Y = 1 \mid S') = 1/2$**

De las probabilidades obtenidas, al aplicar la Definición1 tenemos que :

- $p(Xi = xi, S'i = s'i) = p(X_2 = 0, S' = (x_1 = 0, x_3 = 0, x_5 = 0) = 1/16 > 0$

- $p(Y = y|Xi = xi, S'i = s'i) = p(Y = y|Si = si) \longrightarrow p(Y = 1|X_1 = 0, S') = p(Y = 1|S') \longrightarrow$
  $\longrightarrow 0 \neq 1/2$

Por lo tanto, ***Definición2 == True;***

3.2.3. Variable de entrada X_3

La variable X_3 por enunciado debería de resultar ser relevante débil. Haciendo uso de los algoritmos A_3 primero para descartar que se trata de una variable de relevancia fuerte y A_4 después, se puede comprobar el resultado de relevancia débil de la variable X_3. Igual que con la variable anterior(X_2) se puede hacer una aproximación al problema de forma intuitiva y visual sobre la Tabla básica de entradas de instancias.

Tabla 5: Relevancia de variable X_3

Valores entrada
$X_3 = 0$; $S' = S - \{X_4\} \cong S' = (x_1 = 0, x_2 = 0, x_5 = 0)$; $y = 1$
Cálculo de probabilidades
$p(X_3 = 0, S') = 1/16 > 0$ **$p(Y = 1 \mid X_3 = 0, S') = 0$** **$p(Y = 1 \mid S') = 1/2$**

De las probabilidades obtenidas, al aplicar la Definición1 tenemos que :

- $p(Xi = xi, S'i = s'i) = p(X_3 = 0, S' = (x_1 = 0, x_2 = 0, x_5 = 0) = 1/16 > 0$

- $p(Y = y|Xi = xi, S'i = s'i) = p(Y = y|Si = si) \longrightarrow p(Y = 1|X_3 = 0, S') = p(Y = 1|S') \longrightarrow$
  $\longrightarrow 0 \neq 1/2$

Por lo tanto, ***Definición2 == True;***

3.2.4. Variable de entrada X_4

La variable X_4 por enunciado debería de resultar ser relevante débil. Haciendo uso de los algoritmos A_3 primero para descartar que se trata de una variable de relevancia fuerte y A_4 después, se puede comprobar el resultado de relevancia débil de la variable X_4. Sin embargo, también puede hacerse un cálculo intuitivo para una demostración semiformal de la relevancia de la variable. La variable X_4 está correlacionada con X_2 y X_3, ya que:

$$X_4 = X_2 \oplus X_3.$$

Sabiendo esto de antemano, de forma intuitiva se van a buscar subconjuntos sin estas 2 variables. Así, el subconjunto de datos utilizado para comprobar si es relevante fuerte con respecto a nuestra variable objetivo es*: $S' = S - \{X_2, X_3\}$*

Tabla 6: Relevancia de variable X_4

Valores entrada
$X_4 = 0$; $S' = S - \{X_2, X_3\} \cong S' = (x_1 = 0, x_5 = 0)$; $y = 1$
Cálculo de probabilidades
$p(X_4 = 0, S') = 1/16 > 0$ **$p(Y = 1 \mid X_4 = 0, S') = 0$** **$p(Y = 1 \mid S') = 1/2$**

De las probabilidades obtenidas, al aplicar la Definición1 tenemos que :

- $p(Xi = xi, S'i = s'i) = p(X_4 = 0, S' = (x_1 = 0, x_5 = 0) = 1/16 > 0$

- $p(Y = y|Xi = xi, S'i = s'i) = p(Y = y|Si = si) \longrightarrow p(Y = 1|X_4 = 0, S') = p(Y = 1|S') \longrightarrow$
  $\longrightarrow 0 \neq 1/2$

Por lo tanto, ***Definición2 == True;***

3.2.5. Variable de entrada X_5

La variable X_5 por enunciado debería de resultar ser Irrelevante. Haciendo uso de los algoritmos A_3 primero para descartar que se trata de una variable de relevancia fuerte y A_4 después, se puede comprobar el resultado de Irrelevante para X_5. Para esta variable, de forma intuitiva es muy difícil llegar a un acercamiento a priori, además de que habría que ir probando todas las combinaciones de subgrupos para llegar a alguna conclusión.

Por lo tanto, ***Definición3 == True;***

3.2.6. Resultados obtenidos de relevancia de variables

A continuación, se presentan los resultados de aplicar los algoritmos a cada variable del dataset de entrada. Según los resultados obtenidos, estos coinciden con la relevancia propuesta en el enunciado de la práctica.

Tabla 7: Relevancia de las variables del dataset de entrada

Variable	Relevancia
$\mathbf{X_1}$	*Fuerte*
$\mathbf{X_2}$	*Débil*
$\mathbf{X_3}$	*Débil*
$\mathbf{X_4}$	*Débil*
$\mathbf{X_5}$	*Irrelevante*

Los resultados pueden comprobarse ejecutando el archivo implementado en Java adjunto al documento (ImplementacionAlgoritmos.rar) que realiza de forma automática la determinación de relevancia de las variables del dataset.

```
X1 tiene Relevancia Fuerte? true
X2 Debil en grupo: [1, 3, 5]
 tiene Relevancia Debil? true
X3 Debil en grupo: [1, 2, 5]
 tiene Relevancia Debil? true
X4 Debil en grupo: [1, 3, 5]
 tiene Relevancia Debil? true
X5 tiene Relevancia Debil? false
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ilustración 2: Aplicación que determina la relevancia de las variables de forma automática aplicando los algoritmos mencionados en la Tabla2

4. Justificación de las técnicas de ranking+filtrado

La selección de funciones es muy importante en el ML principalmente porque sirve como una técnica fundamental para dirigir el uso de variables hacia lo que es más eficiente y efectivo, investigando la importancia de cada una de las entradas/variables. Existen 3 métodos principales de selección de características.

Ilustración 3: Métodos existentes de selección de características

1. **Métodos de Filtrado:** Los métodos de filtro se basan en la singularidad general de los datos y seleccionarán un subconjunto de funciones, sin incluir ningún algoritmo de minería de datos. El método de filtro utiliza el criterio de evaluación exacto que incluye distancia, información, consistencia y dependencia. Así se determina la correlación de las características con la variable de resultado.

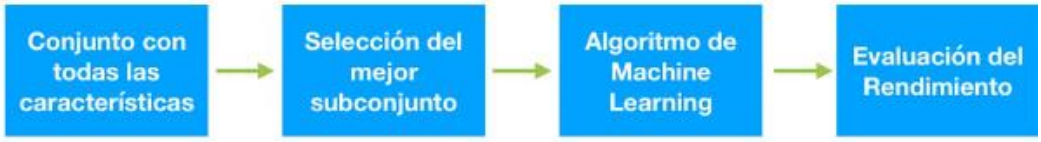

Ilustración 4: Funcionamiento del método de filtrado

2. **Métodos Wrapper:** Esta metodología hace uso de algoritmos de ML para la evaluación a través del rendimiento obtenido. Se busca aquella característica destacada para el algoritmo con el objetivo de mejorar el rendimiento. Se busca un subconjunto de características para entrenar un modelo con ellos, basado en las inferencias extraídas del modelo anterior, se decide agregar o eliminar características de su subconjunto. Así se orienta el problema como un problema de búsqueda. Computacionalmente son métodos muy costosos de implementar.

Ilustración 5: Funcionamiento del método wrapper

3. **Métodos Integrados:** Estos métodos combinan los dos métodos mencionados anteriormente: filtro y "wrapper". La implementación se realiza usando sus propios métodos de selección de características incorporados. Como ejemplos conocidos de estos métodos son la regresión LASSO y RIDGE, con funciones de penalización incorporadas para reducir el sobreajuste. Es menos costoso desde el punto de vista computacional que el método de envoltura y menos propenso al sobreajuste.
4. **Diferencias entre métodos de filtro y envoltura:**
    a. Los métodos de filtrado no aplican un modelo de Aprendizaje Automático (ML) para determinar si una característica es buena o mala, mientras que métodos de envoltura si lo utilizan.
    b. Los métodos de filtrado son más rápidos y eficientes (menos costosos computacionalmente), lo que les hace más indicados para datos masivos.
    c. Los métodos de envoltura siempre pueden proporcionar el mejor subconjunto de características debido a su naturaleza exhaustiva, incluso en situaciones en las que no hay suficientes datos para modelar la correlación estadística de las características.
    d. El uso de características de los métodos de envoltura en su modelo final de ML puede llevar a un ajuste excesivo. En cambio, las características de los métodos de filtro no conducirán a sobreajustes en la mayoría de los casos.

	Filter	***Wrapper***
Computational time	Simple and fast	Complex and Slower
In terms of attribute dependencies	Only to Some degree	Fully incorporated
Cost	Less expensive	Expensive
Scaling ability to high dimensional dataset	Easy	Complex

Ilustración 6: Comparación entre el enfoque de filtro y Wrapper o envoltura(obtenido de [8]*)*

4.1. CfsSubsetEval

Elabora un ranking de subconjuntos de atributos de acuerdo con su correlación basada en la evaluación de una función de evaluación heurística. Para la evaluación de atributos[9] considera la habilidad particular de cada característica junto con el grado de redundancia entre ellos. Los subconjuntos de características que están altamente correlacionados con la clase mientras tienen poca intercorrelación con otros atributos son los preferidos. Se desprende de esta definición, que los atributos irrelevantes serán despreciados por su baja correlación con la clase. La información redundante por otra parte será penalizada ya que el atributo redundante tendrá una alta correlación con una o varias de las características restantes. La inclusión de una característica por tanto depende de si esta es capaz de explicar la clase en fragmentos del espacio de instancias que no han sido ya explicadas por otros atributos. La función de evaluación que utiliza Weka es la siguiente:

$$M_s = \frac{k\bar{r}_{cf}}{\sqrt{k+k(k-1)\bar{r}_{ff}}} \qquad \text{(Ecuación1)}$$

Donde M_s es el mérito heurístico del subconjunto S conteniendo k características, $\overline{r_{cf}}$ es el valor la correlación media entre la clase y la característica f (f ⊂ S) y $\overline{r_{ff}}$ es la mejor correlación entre dos características del conjunto S. CFS simplifica la tarea al suponer que las características son condicionalmente independientes dada la clase, lo que en los casos que existe una fuerte interacción entre distintos atributos, entonces CFS no puede garantizar[10] que los atributos seleccionados sean relevantes.

4.2. ConsistencySubSetEval

Este método de filtrado([11], [12]) busca encontrar el subconjunto de características más grande posible que es consistente. Es más entendible definiendo el contrario, la ***inconsistencia***: se considera un subconjunto de características "inconsistente" si dos instancias comparten todos los mismos valores para las características dadas, pero difieren en sus variables de clase. Matemáticamente, la inconsistencia de un subconjunto de características se encuentra al pasar primero por el conjunto de datos y encontrar todos los patrones únicos producidos por la máscara de función aplicada.

El método hace uso de una aproximación de probabilidad que se basa en el algoritmo LVF que fue propuesto por [13]. La fórmula matemática es la siguiente:

$$Consistency_S = 1 - \frac{\sum_{i=0}^{j} |Di| - |M_i|}{N} \qquad \text{(Ecuación2)}$$

En la Ecuación2, la variable *S* es el subconjunto de atributos del dataset de entrada, y J el número de combinaciones distintas que los atributos generan en *S*. $|D_i|$ consiste en el número de ocurrencias de la i-ésima combinación de atributos, mientras que $|M_i|$ sería la carnalidad de la clase mayoritaria para para la i-ésima combinación de atributos. La variable *N* es el número total de instancias del dataset de entrada.

Este evaluador tiene como objetivo retener la capacidad de discriminación de los datos contenidos en el dataset, y no tanto el maximizar la separabilidad de la clase. Así, se puede formalizar la manera de encontrar el conjunto menor que permite distinguir la clase a partir del conjunto original de entrada. Esto significa que, en un conjunto consistente de características, no hay dos casos cuya tupla de atributos sea la misma que se etiqueten con diferentes etiquetas de clase.

4.3. InfoGainAttributeEval

En estadística matemática , la divergencia de Kullback-Leibler, (también llamada entropía relativa), es una medida de cómo una distribución de probabilidad es diferente de una segunda distribución de probabilidad de referencia. [1] [2] Las aplicaciones incluyen la caracterización de la entropía relativa (Shannon) en los sistemas de información, la aleatoriedad en series de tiempo continuas y la ganancia de información al comparar modelos estadísticos de inferencia. A diferencia de la variación de la información , es una medida asimétrica de distribución y, por lo tanto, no califica como métrica estadística.

Este método evalúa el valor de un atributo midiendo la ganancia de información con respecto a la clase.

$$InfoGain(Clase, Atributo) = H(Clase) - H(Clase \mid Atributo) \qquad \text{(Ecuación3)}$$

Realiza la medición de la contribución de una característica al decremento de la entropía general. De este modo, se entiende que el buen atributo es el que contiene más información y reduce la entropía al máximo.

Para el dataset que tenemos de trabajo, se sabe que son 100 entradas con 16 inputs únicos. La aleatoriedad de las 84 entradas restantes, puede hacer que el peso de las variables sea modificado a variables con menos relevancia por el mero hecho de la aleatoriedad. Esto puede hacer que este método no obtenga el mejor resultado deseado.

4.4. PrincipalComponents o PCA

Este método ([14], [15]) tiene como objetivo reducir el número de variables originales (X1, X2,..., Xn) a un número menor de variables (CP1, CP2,...,CPp), denominadas componentes principales, los cuales son una combinación lineal de las variables iniciales y sintetizan la mayor parte de la información contenida en los datos originales (varianza total o Inercia). PCA transforma n vectores (x_1..., x_d) en un espacio d-dimensional en vectores de la forma (x_1, ..., $x_{d'}$) en un espacio de *d'* dimensiones de la forma:

$$x_i^{'} = \sum_{k=1}^{d'} a_{k,i} e_k \, , d \leq d' \qquad \text{(Ecuación4)}$$

En la fórmula se observa que e_k son los vectores propios correspondientes a los *d'* mayores autovalores para la matriz de covarianzas y $a_{i,k}$ son las proyecciones de los vectores originales x_i en la basa construida con los vectores propios calculados (e_k).

Principal Component Analysis pertenece a la familia de técnicas conocida como "Unsupervised learning", ya que el método no tiene en cuenta la información de la variable de salida o clase. El método de PCA permite por lo tanto "condensar" la información aportada por múltiples variables en solo unas pocas componentes. Esto lo convierte en un método muy útil de aplicar previa utilización de otras técnicas estadísticas tales como regresión, clustering... Aun así, no hay que olvidar que sigue siendo necesario disponer del valor de las variables originales para calcular las componentes.

4.5. WrapperSubsetEval

Se deduce del nombre que se trata de un método de tipo wrapper. Este tipo de métodos utilizan métodos de inducción para la evaluación de los subconjuntos de atributos para su selección. Para este objetivo utiliza algoritmos de clasificación. Por lo tanto, el rendimiento del método está ligado al rendimiento del clasificador seleccionado. Para su implementación, se tiene que definir: (i) cómo se construye el espacio de todos los posibles subconjuntos de atributos. (ii) cómo se mide el rendimiento de aprendizaje para guiar la búsqueda y detenerla alcanzados ciertos criterios. (iii) que método de búsqueda utilizar, teniendo en cuenta que esta elección condicionará la complejidad computacional de la selección que podría volverse intratable al aumentar el número de atributos considerados.

Para esta práctica, se va a utilizar el algoritmo de clasificación *BayesNet*[16]. El comportamiento de este algoritmo depende en gran medida del número de padres que se escojan en su algoritmo de búsqueda. Por defecto hace uso del denominado algoritmo de búsqueda K2, que hace uso de un algoritmo de escalada. Estos algoritmos se caracterizan por encontrar una solución óptima, aunque no sea la mejor. La selección del número de padres es lo que determina el rendimiento de este algoritmo. Elegir sólo 1 o 2 padres[17] serían pocos, convirtiendo a la red en un clasificador ingenuo(1 padre) o semi-ingenuo(2 padres). El nombre para las redes con 2 padres es de TAN, Tree Augmented Naive Bayes, red que relaja la suposición de independencia de las variables utilizando una estructura de árbol, en la que cada característica solo depende de la clase y también de otra característica. Aumentando el número de padres a más de 2, se obtendrán las denominadas redes BAN, Bayes Net Augmented Bayes Network, creando grafos arbitrarios

de expansión de las redes TAN. A la hora de seleccionar el número de padres, se irán haciendo pruebas además de leer documentación científica que nos oriente en este sentido([18], [19], [20], [21]).

5. Resumen de los resultados obtenidos

a. Prueba 1: CfsSubsetVal con búsqueda BestFirst

Para la primera prueba hacemos uso del filtro CfsSubsetVal con el método de búsqueda BestFirst, y seleccionando en el apartado de *modo de selección* todo el conjunto de entrenamiento. Se ha optado por esta opción debido al pequeño número de entradas del dataset, que son 100 inputs. Los resultados obtenidos con este modelo han sido:

=== Run information ===

Evaluator: weka.attributeSelection.CfsSubsetEval -P 1 -E 1

Search: weka.attributeSelection.BestFirst -D 1 -N 5

Relation: experimento1

Instances: 100

Attributes: 6: X1, X2, X3, X4. X5, classY

Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 16

Merit of best subset found: 0.012

Attribute Subset Evaluator (supervised, Class (nominal): 6 classY):

CFS Subset Evaluator

Including locally predictive attributes

Selected attributes: 2,3 : 2----------→ X2, X3

Como puede observarse en los atributos seleccionados por el filtro (X2 y X3), el resultado no es que sea todo lo bueno deseable, ya que se sabe de antemano que la mejor variable es la X1, y esta no ha sido seleccionada. El resultado puede ser debido a que el filtro evalúa el grado de correlación, por lo que se tendrían mejores resultados si las variables estuvieran relacionadas con la clase en vez de entre sí.

Debido al mal resultado, se ha modificado el modo de selección de atributos aplicando el ***cross validation 10x1.*** De este modo se particiona el dataset en 19 conjuntos, 9 de entrenamiento y 1 de validación. Los resultados obtenidos son:

=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

number of folds (%) attribute

4(40 %) 1 X1

8(80 %) 2 X2

8(80 %) 3 X3

6(60 %) 4 X4

0(0 %) 5 X5

Se puede observar que los resultados han mejorado, y que ha dado un porcentaje de importancia mayor a las variables que en el caso anterior no habían sido seleccionadas como X1 y X4. Sin embargo, el porcentaje de importancia de X1 sigue por debajo del resto, algo que no es correcto por definición. La variable X5 queda relegada sin tener ningún peso de importancia en el dataset.

b. Prueba 2: ConsistencySubSetEval con búsqueda BestFirst

Este filtro hace énfasis formalizar la manera de encontrar el conjunto menor que permite distinguir la clase a partir del conjunto original de entrada. El objetivo es buscar el subconjunto de variables que de forma consistente sean capaces de definir la salida. Para el experimento se ha usado todo el dataset, obteniendo los siguientes resultados:

=== Attribute Selection on all input data ===

Search Method:

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 16

Merit of best subset found: 1

Attribute Subset Evaluator (supervised, Class (nominal): 6 classY):

Consistency Subset Evaluator

Selected attributes: 1,2,3 : 3 ----------→X1, X2, X3

Este filtro ha hilado más fino que el anterior, habiendo seleccionado las variables principales del dataset tal y como estaba formulado el ejercicio. La consistencia de este subgrupo de variables con las salidas es clara, ya que : $\mathbf{Y= X_1 \oplus X_2 \oplus X_3}$.

c. Prueba 3: InfoGainAttributeEval con búsqueda Ranker

Este filtro trabaja evaluando el valor de un atributo midiendo la ganancia de información con respecto a la clase. Este filtro utiliza el método de búsqueda Ranker, debido a que los atributos son evaluados por separado. Los resultados obtenidos con esta técnica y los parámetros mencionados son:

=== Attribute Selection on all input data ===

Search Method:

Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 6 classY):

Information Gain Ranking Filter

Ranked attributes:

0.010244 3 X3

0.007228 2 X2

0.002674 1 X1

0.00123 4 X4

0.000339 5 X5

Selected attributes: 3,2,1,4,5 : 5

Se observa que la valoración realizada con este algoritmo, no ha sido óptima, ya que valora con mayor ganancia de información a X3 y X2 por encima de X1. Modificando el modelo de evaluación a ***cross validation 10x1***, los resultados obtenidos son los mismos. A la hora de analizar los resultados , también hay que tener en cuenta que de las 16 entradas diferentes que generan el dataset básico, se han creado otras 84 aleatorias, que pueden de alguna forma hacer variar el peso de las variables de entrada.

Como aspecto positivo, se puede decir que las 3 variables con mayor ganancia de información son las que más correlacionadas están con la variable de salida, aunque el orden no sea el que la relevancia les presume.

d. Prueba 4: Algoritmo PrincipalComponents (PCA) con método de búsqueda Ranker

La idea de este algoritmo es la de buscar dentro de un conjunto de datos con un número alto de variables, la existencia de correlación entre las variables y la definición de nuevos atributos artificiales partir de las anteriores. Las nuevas variables son referidas como Componentes Principales: i) el Primer Componente Principal, la componente más importante, captura el porcentaje de varianza mayor del conjunto de datos original ii) el Segundo Componente Principal, es la segunda componente más grande iii) así sucesivamente para el resto de las componentes.

Aplicando esta técnica a nuestro dataset, los resultados obtenidos han sido los siguientes:

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (unsupervised):
        Principal Components Attribute Transformer

Correlation matrix
  1      0.12   0.1    0.02   0.04
  0.12   1     -0.06   0.06   0.24
  0.1   -0.06   1      0     -0.1
  0.02   0.06   0      1      0.14
  0.04   0.24  -0.1    0.14   1


eigenvalue      proportion      cumulative
  1.35552         0.2711          0.2711        -0.633X5=1-0.607X2=1-0.358X4=1-0.25X1=1+0.201X3=1
  1.10703         0.22141         0.49251       -0.709X3=1-0.686X1=1+0.139X5=1-0.077X2=1-0.034X4=1
  0.97484         0.19497         0.68748       -0.86X4=1+0.331X2=1+0.279X1=1-0.269X3=1-0.026X5=1
  0.82762         0.16552         0.853         0.619X3=1-0.594X1=1+0.316X5=1+0.287X2=1-0.285X4=1
  0.73499         0.147           1             0.693X5=1-0.658X2=1-0.223X4=1+0.191X1=1+0.033X3=1

Eigenvectors
 V1      V2      V3      V4      V5
-0.2498 -0.6858  0.2788 -0.5941  0.1915 X1=1
-0.6073 -0.0771  0.331   0.2875 -0.658  X2=1
 0.2006 -0.7094 -0.269   0.619   0.0331 X3=1
-0.3582 -0.0343 -0.8601 -0.2851 -0.2225 X4=1
-0.6327  0.1393 -0.0261  0.3162  0.6926 X5=1

Ranked attributes:
 0.729  1 -0.633X5=1-0.607X2=1-0.358X4=1-0.25X1=1+0.201X3=1
 0.507  2 -0.709X3=1-0.686X1=1+0.139X5=1-0.077X2=1-0.034X4=1
 0.313  3 -0.86X4=1+0.331X2=1+0.279X1=1-0.269X3=1-0.026X5=1
 0.147  4 0.619X3=1-0.594X1=1+0.316X5=1+0.287X2=1-0.285X4=1
 0      5 0.693X5=1-0.658X2=1-0.223X4=1+0.191X1=1+0.033X3=1

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

Ilustración 7: Resultados obtenidos de aplicar PCA al dataset en Weka

En los resultados obtenidos con Weka, se observa que no se proporciona la relevancia de los atributos originales, sino de los nuevos por orden de varianza de cada uno. Se generan nuevas características de para que luego sea (aparentemente) sencillo realizar la selección de estos. La proyección de las variables originales se realiza de forma lineal sobre los nuevos ejes, seleccionando aquellas con dirección de máxima varianza.

Desde la matriz de correlación se observan el grado de correlación entre las propias variables, algo que es fundamental para que este método obtenga buenos resultados. Los resultados de correlación entre variables son muy pobres, ya que puede observarse que la mayor correlación encontrada es entra la X2 y X5 con un 24%, algo que no responde a las premisas y restricciones del problema. Esto puede deberse a la forma de generar el dataset, que de alguna forma desvirtúa la correlación y restricciones del experimento. Esto podría invalidar el experimento realizado ya que para la aplicación de este método es requisito que exista un alto grado de correlación que sea indicativo de que exista información redundante y, por tanto, con unos pocos factores sea posible explicar gran parte de la variabilidad total.

Según el valor de información que se van generando se ve que en este caso sí que en principio se observa como 3 variables alcanzan casi un 70% de la información (68,748%).

Con estos valores se representan los autovalores (eigenvectors), tras la selección de los componentes principales. Estos vectores generan una matriz con tantas columnas como componentes principales y tantas filas como atributos. La matriz sirve para saber cuánta información es capaz de recoger las componentes del muestreo original y como se distribuye esta información capturada.

Los nuevos componentes generados son medidos en ránking según el valor de importancia de información que acumulan. En principio parece que este método no es el más recomendado para aplicar a este dataset para la reducción de dimensiones debido a que las variables están poco correlacionadas entre sí, y además las variables son de naturaleza no lineal y binaria, lo que genera varianzas artificiales imposible de capturar por PCA.

e. Prueba 5: Método wrapper *WrapperSubsetEval + GreedyStepwise*

Para probar el dataset con este método, se ha seleccionado el método de búsqueda GreedyStepwise. Los métodos wrapper necesitan un modelo de clasificación, que en nuestro caso para el experimento será el BayesNet. Los resultados obtenidos son los siguientes:

=== Attribute Selection on all input data ===

Search Method:

Greedy Stepwise (backwards).

Start set: all attributes

Merit of best subset found: 0.588

Attribute Subset Evaluator (supervised, Class (nominal): 6 classY):

Wrapper Subset Evaluator

Learning scheme: weka.classifiers.bayes.BayesNet

Scheme options: -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Subset evaluation: classification accuracy

Number of folds for accuracy estimation: 5

Selected attributes: 1,2,3 : 3 ----------→ X1, X2, X3

Los resultados obtenidos como pueden observarse son coherentes con el enunciado y las restricciones del enunciado. Se ha utilizado el método wrapper con la opción de búsqueda hacia atrás, ya que por defecto con la búsqueda hacia adelante los resultados obtenidos eran muy pobres:

=== Attribute Selection on all input data ===

Search Method:

Greedy Stepwise (forwards).

Start set: no attributes

Merit of best subset found: 0.55

Attribute Subset Evaluator (supervised, Class (nominal): 6 classY):

Wrapper Subset Evaluator

Learning scheme: weka.classifiers.bayes.BayesNet

Scheme options: -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Subset evaluation: classification accuracy

Number of folds for accuracy estimation: 5

Selected attributes: 2 : 1 ----------→ X2

Por otro lado, mencionar las diferencias del wrapper usando BayesNet con diferentes padres. El resultado más acorde con el enunciado de la práctica ha sido para P -1, es decir con 1 sólo padre o clasificador ingenuo. En la siguiente Tabla podemos observar los resultados:

Tabla 8: Resultados de aplicar el método wrapper según el número de padres

Número de padres	*1*	*2*	*3*	*4*
Variables resultado	X1, X2, X3	X1, X4	X1, X4	X1, X4

Aplicando el modelo de evaluación a ***cross validation 10x1***:

Tabla 9: Resultados de aplicar el método wrapper según el número de padres con un modelo de evaluación cross validation 10x1

Número de padres	*1*	*2*	*3*	*4*
Variables resultado	Nº of folds (%) attr. 10(100 %) 1 X1 10(100 %) 2 X2 9(90 %) 3 X3 2(20 %) 4 X4 1(10 %) 5 X5	Nº of folds (%) attr. 10(100 %) 1 X1 0(0 %) 2 X2 0(0 %) 3 X3 10(100 %) 4 X4 0(0 %) 5 X5	Nº of folds (%) attr. 10(100 %) 1 X1 0(0 %) 2 X2 0(0 %) 3 X3 10(100 %) 4 X4 0(0 %) 5 X5	Nº of folds (%) attr. 10(100 %) 1 X1 0(0 %) 2 X2 0(0 %) 3 X3 10(100 %) 4 X4 0(0 %) 5 X5

Se observa que los resultados se asemejan al de seleccionar el conjunto de datos de forma total, y que además se repiten los resultados con el número de padres: 2,3 y 4.

A modo de conclusión para este método, podemos comentar que las variables seleccionadas por el clasificador serían X1, X2 y X3 de la primera prueba realizada, y que están en línea con el enunciado y los resultados obtenidos con los otros métodos. En principio puede parecer un tanto contradictorio que los mejores resultados se hayan obtenido con el clasificador ingenuo frente al resto que se supone más precisos.

5.1. Tabla de resultados global

A continuación, se muestra la tabla con los resultados de los experimentos. Hay que comentar que el orden de las variables seleccionadas es irrelevante salvo en el caso del método de búsqueda Ranker, que califica las variables en orden de importancia según el método.

Tabla 10: Resultados de los diferentes métodos aplicados al dataset

Método	Tipo	Opciones	Búsqueda	Variables
CfsSubsetEval	Filtro	Defecto	BestFirst	X2, X3
ConsistencySubsetEval	Filtro	Defecto	BestFirst	X1, X2, X3
InfoGainAttributeEval	Filtro	Defecto	Ranker	X3, X2, X1, X4, X5
PCA[1]	Filtro	Defecto	Ranker	X1, X2, X3, X4, X5
WrapperSubsetEval	Wrapper	BayesNet, K2, 1 padre, Backwards	GreedySetpwise	X1, X2, X3

[1] Los resultados de este método se explican con detalle en Prueba 4

6. Explicación y análisis de los resultados obtenidos

En principio comentar ciertas limitaciones de práctica, que básicamente hace referencia al dataset. Se trata de un dataset con pocas entradas, sólo 100, y además sólo 16 diferentes. Además, el resto se crean de forma aleatoria replicando las instancias diferentes, lo cual puede crear datasets que de alguna forma desvirtúen el peso real que las variables tienen en la clase o salida. Pienso que esta ha sido la limitación principal para la obtención de resultados.

Por otro lado, parece que en general, los métodos utilizados han respondido bien y las premisas del enunciado han sido bien interpretadas por éstos. La variable más relevante a priori, no ha sido tan refrendado por los métodos, pero en el caso de la variable irrelevante, sí que ha sido unánime que en ninguno ha sido seleccionada, por lo que estos métodos por lo menos han demostrado ser efectivos con las variables prescindibles. Esto concuerda con el objetivo inicial de aplicar estos métodos a la aplicación de ML cuando estamos trabajando con datasets en una aplicación real, para de entrada eliminar las variables innecesarias y conseguir así mejor eficiencia algorítmica eliminando el ruido que estas variables puedan generar.

Al aplicar los métodos de selección de variables, se ha observado que al aplicar modo de selección de atributos con cross validation 10x1, los resultados obtenidos en general eran óptimos y se aproximaban más a lo que se sabía por el enunciado.

Los métodos que para este dataset quizás se hayan mostrado más débiles han sido los filtros CfsSubsetEval, InfoGainAttributeEval y PCA. Todos estos métodos son filtros, y por las características del dataset han obtenido los resultados más pobres. Para el método PCA, hay que comentar que el dataset mostraba una serie de características que claramente perjudicaban el trabajo de este. Las variables del dataset se observa que tienen poca correlación entre sí, y además son de naturaleza no lineal y binaria, lo que condiciona y se generan varianzas artificiales que son imposible de capturar por el algoritmo.

El método wrapper se ha mostrado robusto, y además ofrece la posibilidad de aplicar muchos cambios de parámetros que optimicen los resultados. En nuestro caso, por defecto el método Forward, método que comienza con un modelo nulo y luego empieza a ajustar el modelo con cada característica individual de una en una y seleccionando la característica con el valor p mínimo. Al aplicarlo al problema, los resultados han sido pobres, sin embargo, al aplicar el modelo hacia atrás (Backward), el wrapper ha respondido como se esperaba según los datos. El método Backward, comienza con el modelo completo (incluidas todas las variables independientes) y luego elimina la característica insignificante con el valor p más alto (> nivel de significación). Este proceso se repite una y otra vez hasta que se tiene el conjunto final de características significativas. Para nuestro dataset este modelo ha resultado más eficiente. El éxito de este modelo puede deberse a dos factores i) por la visión global que ofrecen al considerar subconjuntos de atributos, evaluando varios atributos de forma concurrente. Así se analiza la influencia de cada atributo en la clase y también su relación con el resto de las variables ii) por la integración un algoritmo de clasificación desde el inicio para poder utilizar técnicas testadas previamente y que son conocidas su complejidad y rendimiento para el proceso de selección mismo

7. Referencias

[1] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997, doi: https://doi.org/10.1016/S0004-3702(97)00043-X.

[2] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[3] D. Galles and J. Pearl, "Axioms of causal relevance," *Artif. Intell.*, vol. 97, no. 1, pp. 9–43, 1997, doi: https://doi.org/10.1016/S0004-3702(97)00047-7.

[4] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, no. 1, pp. 245–271, 1997, doi: https://doi.org/10.1016/S0004-3702(97)00063-5.

[5] Y. Mao and Y. Yang, "A Wrapper Feature Subset Selection Method Based on Randomized Search and Multilayer Structure," *Biomed Res. Int.*, vol. 2019, p. 9864213, 2019, doi: 10.1155/2019/9864213.

[6] S. Das, "Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 74–81.

[7] N. El Aboudi and L. Benhlima, "Review on wrapper feature selection approaches," in *2016 International Conference on Engineering & MIS (ICEMIS)*, 2016, pp. 1–5, doi: 10.1109/ICEMIS.2016.7745366.

[8] D. Gnanambal, D. Thangaraj, Meenatchi V T, and D. Gayathri, "Classification Algorithms with Attribute Selection: an evaluation study using WEKA," *Int. J. Adv. Netw. Appl.*, vol. 09, no. 06, pp. 3640–3644, 2018.

[9] M. A. Amrita, "Performance Analysis Of Different Feature Selection Methods In Intrusion Detection," *Int. J. Sci. Technol. Res.*, 2013, Accessed: May 27, 2022. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.307.2033.

[10] M. A. Hall and L. A. Smith, "Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper," in *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, 1999, pp. 235–239.

[11] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artif. Intell.*, vol. 151, no. 1, pp. 155–176, 2003, doi: https://doi.org/10.1016/S0004-3702(03)00079-1.

[12] K. Shin and X. M. Xu, "Consistency-Based Feature Selection BT - Knowledge-Based and Intelligent Information and Engineering Systems," 2009, pp. 342–350.

[13] H. Liu and R. Setiono, "A Probabilistic Approach to Feature Selection - A Filter Solution," 1996.

[14] L. I. Smith, "A tutorial on Principal Components Analysis Chapter 1," 2002.

[15] M. Ringnér, "What is principal component analysis?," *Nat. Biotechnol.*, vol. 26, no. 3, pp. 303–304, 2008, doi: 10.1038/nbt0308-303.

[16] T. Bayes, "LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S," *Philos. Trans. R. Soc. London*, pp. 370–418.

[17] P. Larrañaga, "Clasificadores Bayesianos," Universidad del País Vasco–Euskal Herriko Unibertsitatea.

[18] L. E. Sucar, "Redes Bayesianas," Sta. María Tonantzintla, Puebla, 72840, México. Accessed: May 28, 2022. [Online]. Available: https://ccc.inaoep.mx/~esucar/Clases-mgp/caprb.pdf.

[19] R. G. Brereton, "Introduction to Bayesian methods," *J. Chemom.*, vol. 36, 2021.

[20] C. L. de C. Vásquez, "Redes bayesianas con algoritmos basados en restricciones, scores e híbridos aplicados al problema de clasificación," *An. Científicos*, vol. 80, no. 1, pp. 15–25, Jun. 2019, doi: 10.21704/AC.V80I1.1370.

[21] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 462–467, 1968, doi: 10.1109/TIT.1968.1054142.