# Generación Automática de Explicaciones en Lenguaje Natural para Árboles de Decisión de Clasificación

B. López-Trigo, Jose M. Alonso, A. Bugarín
Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS),
Universidade de Santiago de Compostela,
Campus Vida, E-15782, Santiago de Compostela, Spain
Email: bruno.lopez.trigo@rai.usc.es, {josemaria.alonso.moral, alberto.bugarin.diz}@usc.es

***Resumen*—En este trabajo describimos un modelo de explicaciones en lenguaje natural para árboles de decisión para clasificación. Las explicaciones incluyen aspectos globales del clasificador y aspectos locales de la clasificación de una instancia concreta. La propuesta está implementada en el servicio Web de código abierto ExpliClas [1], que en su versión actual opera sobre árboles construidos con Weka y conjuntos de datos con atributos numéricos. Ilustramos la viabilidad de la propuesta con dos casos de ejemplo, donde mostramos paso a paso cómo el modelo explica los respectivos árboles de clasificación.**

***Index Terms*—Explicabilidad, Soft Computing, Árboles de decisión para Clasificación, Generación de Lenguaje Natural**

## I. Introducción

La generalización del uso de las nuevas tecnologías ha hecho que hoy trabajemos y vivamos rodeados de sistemas inteligentes [2]. Términos como ciudad inteligente, fábrica, casa, coche o teléfono inteligentes, son cada vez más populares. En realidad, existen multitud de dispositivos dotados de cierta inteligencia que nos asisten en el día a día, muchas veces sin que seamos totalmente conscientes de ello. Mención especial merece el teléfono móvil, que nos ofrece multitud de aplicaciones casi para cualquier cosa que podamos imaginar y va con nosotros a todas partes. Se puede afirmar que, si bien en el pasado vivimos una revolución industrial, ahora estamos viviendo una revolución social impulsada por la Inteligencia Artificial (IA).

Cuando un sistema inteligente toma decisiones que nos afectan (ej. filtrar llamadas, diagnóstico médico, concesión de un préstamo, etc.), surgen multitud de preguntas que deberíamos hacernos [3]: ¿quién es el responsable de las consecuencias colaterales que pudieran derivarse de las decisiones tomadas? ¿cuáles son las consecuencias éticas? ¿puede haber consecuencias legales?

Desde el punto de vista legal, el Parlamento Europeo aprobó una nueva Regulación General de Protección de Datos [4] que entró en vigor el 25 de mayo de 2018. La nueva regulación enfatiza el derecho de los ciudadanos a pedir explicaciones, independientemente de que las decisiones que les afectan sean tomadas por una persona o un programa informático. Esto significa que los ciudadanos pueden pedir a las empresas que les den explicaciones asociadas a las decisiones tomadas por los sistemas inteligentes que utilizan.

Desde un punto de vista técnico: ¿puede explicarnos la aplicación que tomó una decisión por qué tomó esa decisión y no otra? Para esto, hay básicamente dos opciones [5]: (1) el sistema inteligente está construido siguiendo un modelo interpretable (también llamado de caja blanca) que un operario experto puede analizar y entender a fin de elaborar una explicación; o (2) el sistema está construido siguiendo un modelo explicable que genera explicaciones por sí mismo. La DARPA planteó en 2016 las siguientes cuestiones técnicas [5]: ¿puede una máquina inteligente aprender de forma autónoma a explicar su comportamiento? ¿está preparada la generación actual de sistemas inteligentes para dar explicaciones de forma clara, sin ambigüedades, tanto a públicos especializados como no especializados? Y lanzó el reto de crear una nueva generación de sistemas inteligentes explicables entre 2017 y 2021. El reto fue lanzado inicialmente a universidades y centros de investigación americanos, con énfasis en la creación de equipos multidisciplinares que abordasen no sólo aspectos algorítmicos sino también de implementación y evaluación con personas. Los equipos seleccionados empezaron a trabajar en mayo de 2017 pero a día de hoy sólo hemos encontrado resultados muy preliminares (ej. [6], [7]).

Hasta donde nosotros sabemos, en la práctica, la responsabilidad de generar explicaciones recae directamente en el operario asociado al sistema inteligente, si está disponible para ello [8]. Aunque hay sistemas basados en conocimiento que son interpretables, en los últimos años son cada vez más populares las técnicas de IA para aprendizaje automático y minería de datos, supervisadas y no supervisadas (es decir, con o sin intervención humana). Estos sistemas se están demostrando ciertamente útiles y versátiles, pero la mayoría no suelen tener ninguna capacidad explicativa ni tampoco pueden ser interpretados fácilmente por personas (en cuyo caso se dice que son sistemas de caja negra).

Por tanto, el nuevo marco legal demanda que los expertos en IA desarrollen nuevos algoritmos que proporcionen explicaciones de forma automática.

En este trabajo, presentamos un modelo para la interpre-

tación de uno de los algoritmos de IA más interpretable, como son los árboles de decisión para clasificación, que introduciremos en la Sección II. El generador de explicaciones basado en dicho modelo y la combinación de técnicas de análisis inteligente de datos y generación de lenguaje natural se describe en la Sección III. La Sección IV presenta 2 casos de uso ilustrativos. Finalmente, la Sección V resume las principales conclusiones y apunta líneas de trabajo futuro.

## II. Clasificación con Árboles de Decisión

Dentro del aprendizaje supervisado a partir de conjuntos de datos, los métodos basados en modelos se caracterizan por representar el conocimiento aprendido en algún formalismo de representación que explicita dicho conocimiento. Una ventaja importante de esta aproximación es que, una vez que se dispone del modelo, éste puede aplicarse directamente sobre nuevas instancias (por ejemplo, en problemas de predicción, como la clasificación) sin necesidad de seguir manteniendo los datos de entrenamiento. Los árboles de decisión utilizan como formalismo de representación un árbol donde los nodos representan condiciones sobre los valores de los atributos del conjunto de datos, que se organizan jerárquicamente, y donde las ramas de cada nodo corresponden a posibles valores del atributo. Hay diferentes métodos inductivos [9], [10] para la construcción de un árbol de decisión, pero todos ellos suelen utilizar estrategias "divide y vencerás" que construyen el árbol desde la raíz a las hojas seleccionando en cada nodo intermedio el atributo y la condición que particiona el conjunto de datos de la mejor manera posible, habitualmente en base a criterios de entropía y de maximización de la ganancia de información [11].

En el caso concreto de los árboles de clasificación, los nodos hojas contienen, idealmente, un conjunto de instancias correspondientes a la misma clase. La aplicación para la clasificación de nuevas instancias se inicia evaluando la condición del nodo raíz para los atributos de dicha instancia y continuando el recorrido por las ramas y nodos correspondientes. El proceso de clasificación finaliza cuando se alcanza un nodo hoja, que indica la clase que corresponde a la instancia. En la práctica, la condición de que un nodo hoja contenga únicamente instancias de la misma clase (nodo "puro") es demasiado restrictiva, con lo que dicha condición se debe relajar dentro de unos márgenes de pureza. Ello da lugar, por otra parte, a que los árboles clasifiquen incorrectamente algunos (idealmente muy pocos) casos, característica que se recoge en la matriz de confusión entre clases.

Nuestro modelo de explicación de árboles de clasificación se basa en estos aspectos que acabamos de comentar. Por un lado, una caracterización global del problema de clasificación y del árbol inducido. Por otro, una explicación del recorrido por el árbol en la tarea de clasificación. Veremos en la siguiente sección estos aspectos en mayor detalle.

## III. Modelo para la Generación de Explicaciones

La generación de texto en Lenguaje Natural (popularmente conocida como NLG por el acrónimo de "Natural Language Generation") constituye una línea de investigación destacada en el área de la IA y la Lingüística Computacional [12].

En este trabajo, tomamos como punto de partida la arquitectura NLG más popular, inicialmente propuesta por Reiter y Dale [13], y la Teoría Computacional de Percepciones propuesta por Zadeh [14]. La generación de explicaciones en Lenguaje Natural se hace combinando plantillas y librerías de código abierto para la realización lingüística [15].

Planteamos la explicación de clasificadores mediante árboles de decisión a dos niveles (global y local), tal y como se describe a continuación. Todos los ejemplos utilizados en las siguientes secciones para ilustrar la propuesta se pueden reproducir mediante el servicio web ExpliClas [1].

### *III-A. Explicación global de un clasificador*

El primer nivel es la explicación que denominamos global, que se orienta a describir el comportamiento general de un árbol de clasificación dado, aprendido a partir de un determinado conjunto de datos. La información que se incluye en la explicación global se refiere esencialmente a características del propio problema de clasificación y su rendimiento. Los datos de entrada para esta explicación provienen del propio conjunto de datos y de la matriz de confusión del clasificador aprendido.

La planificación de la explicación global contiene los elementos que se muestran a continuación:

- **Contextualización del problema**, que enumera las clases del mismo.

  **Prototipo**: `There are [N] types of beer: [Class1], [Class2], ... and [ClassN].`

  **Ejemplo**: `There are 8 types of beer: Blanche, Lager, Pilsner, IPA, Stout, Barleywine, Porter and Belgian Strong Ale.`

- **Fiabilidad del clasificador**, que evalúa el porcentaje global de clasificaciones correctas sobre el conjunto de datos de aprendizaje, incluyendo una valoración cualitativa del mismo de acuerdo con una definición establecida de valores lingísticos.

  **Prototipo**: `This classifier is [very reliable / quite confusing / very confusing] because correctly classified instances represent [percentage]%.`

  **Ejemplo**: `This classifier is very reliable because correctly classified instances represent 94.75%.`

- **Confusión del clasificador**, destacando qué clases se ven afectadas en mayor medida por dicha confusión. Aquí se interpreta la matriz de confusión del clasificador como una matriz de adyacencia de un grafo, cuyos ciclos se entienden como posibles caminos cerrados de confusión entre clases. Se toma el camino de mayor

longitud para ser incluido en la explicación. En caso de que el nivel de confusión sea bajo se omitirá esta parte de la explicación. A la hora de enumerar las clases se busca limitar la longitud de la explicación, tratando de forma diferente los casos en que el camino cerrado de confusión es largo (muchas clases confundidas) o corto (número reducido de clases confundidas) de modo que la longitud de la explicación sea lo más corta posible. Así, en el primer caso, se enumeran las clases para las que no hay confusión (expresándolas como excepciones) y en el segundo caso se enumeran las clases para las que hay confusión. En situaciones intermedias, como la del siguiente ejemplo, se citan los casos concretos.

**Prototipo**: `There may be some confusion among samples related to [a few / some / most / all] types of [object]. But among all of them [the pair / pairs [[class1]; [class2]] and [[classM-1]; [classM]] [is / are] the most confused.`

**Ejemplo**: `There may be some confusion among samples related to some types of beer. But among all of them the pair [IPA; Barleywine] is the most confused.`

- **Confusión elevada entre clases**, donde se destacan aquellos pares de clases que presenten un elevado nivel de confusión y no estén incluidas en los ciclos anteriores. Al igual que en los ejemplos mostrados previamente, se incluye una valoración lingüística además de la numérica.

  **Prototipo**: `[On the one hand / On the other hand], the following pairs are [eventually / often / usually] misleaded: Class [class1] is confused with class [class2] in [percentage]% of cases.`

  **Ejemplo**: `On the one hand, the following pairs are eventually misleaded: class headlamps is confused with class build wind float in 10.34% of cases.`

### *III-B. Explicación local de una instancia*

El segundo nivel es la explicación local, que se orienta a explicar cuál es el resultado de la clasificación obtenida al aplicar el clasificador sobre una nueva instancia. La información que se incluye en la explicación local se refiere al recorrido por el árbol de clasificación desde la raíz hasta una hoja, determinado por las condiciones que se cumplen en los diferentes nodos del árbol para la instancia que se quiere clasificar. La versión actual del modelo que hemos definido para la generación de explicaciones en lenguaje natural, se aplica únicamente a atributos de tipo numérico, lo que nos permite dar una cierta flexibilidad en la explicación, para considerar posibles alternativas a la clasificación real. Para ello incluimos una cierta tolerancia en cuanto a los valores umbral de las condiciones, para de este modo contemplar que se puedan dar pequeñas variaciones en el valor de un atributo, que pudieran derivar en una clasificación diferente. Los datos de entrada para la explicación local son la instancia a clasificar, el árbol de clasificación y el valor de tolerancia permitido (por defecto, 5 % sobre el valor de cada atributo).

La planificación de la explicación local contiene los siguientes elementos:

- **Descripción de la clase**, donde se expresa cuál es el resultado de la clasificación y un resumen lingüístico de los valores de los atributos que han dado lugar a dicha clasificación. En el resumen se incluyen, para cada atributo $X$ expresiones del tipo "$X$ es $A$", donde $A$ es un valor lingüístico predefinido.

  **Prototipo**: `[Object] is type [output class] because its [attribute1] is [lingTerm1Attribute1] ([or [lingTerm2Attribute1]]), its [attribute2] and [attribute3] are [lingTerm1Atributo1...], ... and its [attributeN] is [lingTerm1AttributeN].`

  **Ejemplo**: `Beer is type Porter because its strength is standard and its color is brown.`

- **Explicaciones alternativas**, que se construyen en base al umbral de tolerancia mencionado anteriormente. Se ha establecido un margen de tolerancia del 5 % para cada una de las condiciones nodo que justifican la clasificación, de modo que se exploran y se incluyen en la explicación las posibles clasificaciones alternativas que se obtendrían en caso de que los valores de los atributos cumpliesen las condiciones dentro del margen de tolerancia.

  **Prototipo**: `However, this [object] may be also [alternativeClass1] because its [alternativeAttribute1] is quite close to the split value ([thresholdValue]). For these specific values, it is [unlikely / quite likely / just as likely] to be [alternativeClass1].`

  **Ejemplo**: `However, this beer may be also Stout because its color is quite close to the split value (30.45). For these specific values, it is just as likely to be Stout.`

Por último también se incorporan en la explicación alternativa aquellas clases para las cuales hay un elevado nivel de confusión en general con la clase original. Para ello se tiene en cuenta la matriz de confusión en lo que respecta a las clases implicadas, adoptando por tanto una cierta perspectiva global. Así, si las clases tienen, en general, un

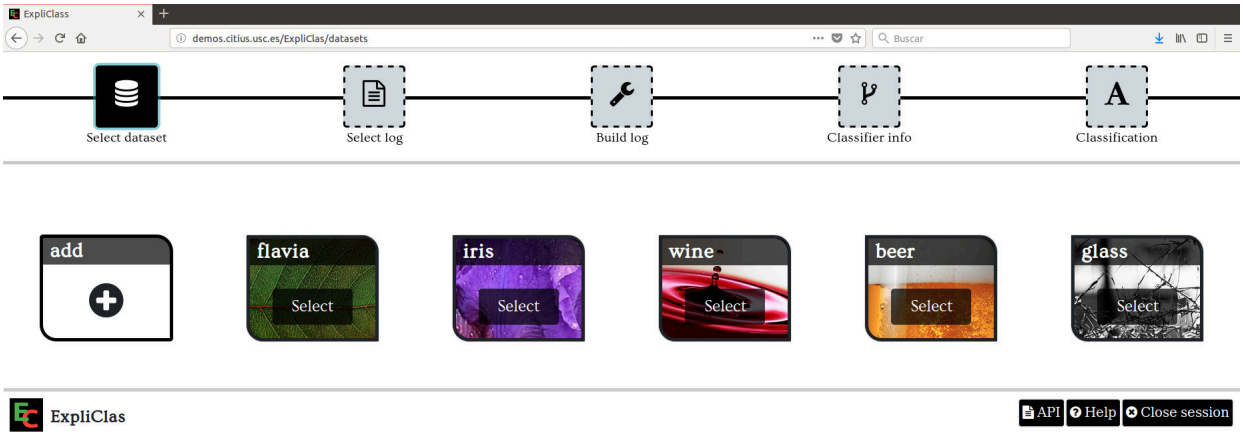


Figura 1. Página de inicio del Servicio Web ExpliClas [1].

elevado nivel de confusión, la explicación enfatiza este aspecto, mientras que si el nivel de confusión es bajo se presentará como un caso de cierta excepción. Mostramos a continuación un ejemplo de esta última situación:

**Prototipo**: `But [alternativeClass1] will be an exception because class [outputClass] is confused with [alternativeClass1] only in [percentage]% of cases.`

**Ejemplo**: `But Stout will be an exception because class Porter is confused with Stout only in 2% of cases.`

## IV. Algunos Casos de Uso

Una vez descritos los elementos que componen cada explicación, veremos en esta sección dos ejemplos completos, con los que ilustraremos el funcionamiento de nuestra propuesta paso a paso. En ambos casos se aprenden clasificadores utilizando el algoritmo C4.5 [10], en la implementación disponible en Weka (J48) [16], [17]. Tanto los dos ejemplos mostrado (IRIS y FLAVIA), como otros disponibles, se pueden reproducir con el servicio Web ExpliClas [1] (Fig. 1).

### *IV-A. Conjunto de datos IRIS*

El conjunto de datos IRIS (uno de los más conocidos del repositorio [18]) está formado por 150 instancias, 4 atributos numéricos y 3 clases. El árbol de clasificación generado por Weka (Fig. 2) está formado por 9 nodos totales, 5 de ellos nodos-hoja que deciden la clasificación y los 4 nodos restantes con las condiciones (comparaciones sobre los valores de los atributos) para decidir la clasificación. Se trata, por tanto, de un árbol simple que utilizaremos como primer ejemplo.

La explicación global generada en este caso es la siguiente:

```
There are 3 types of iris:
Setosa, Virginica and Versicolor.
This classifier is very reliable
because correctly classified
instances represent 96%.
```

```
Petal-Width <= 0.6: 1.0 (50.0)
Petal-Width > 0.6
|   Petal-Width <= 1.7
|   |   Petal-Length <= 4.9: 2.0 (48.0/1.0)
|   |   Petal-Length > 4.9
|   |   |   Petal-Width <= 1.5: 3.0 (3.0)
|   |   |   Petal-Width > 1.5: 2.0 (3.0/1.0)
|   Petal-Width > 1.7: 3.0 (46.0/1.0)
```

Figura 2. Árbol de clasificación correspondiente al conjunto de datos IRIS (captura de pantalla de Weka [17]).

La explicación local, para la instancia de la Fig. 3 (Sepal-Length: 5.6, Sepal-Width: 3, Petal-Length: 4.1, Petal-Width: 1.3) es la siguiente:

```
Iris is type Virginica because
its petal-length and petal-width
are medium.
```

En este caso, la explicación consiste en indicar los valores lingüísticos correspondientes a los valores numéricos de los atributos que han dado lugar a la clasificación, tal y como se detalla en la figura.

Sin embargo, si tomamos una instancia cuyos valores sean precisamente los de umbrales de los nodos intermedios (Sepal-Length: 5.6, Sepal-Width: 3, **Petal-Length: 4.9**, **Petal-Width: 0.6**), la explicación resulta más extensa:

```
Iris is type Setosa because its
petal-width is low.
However, this iris may be also
Virginica because its petal-width
is quite close to the split value
(0.6).
It may be also Versicolor because
its petal-width and petal-length
are quite close to the split values
(0.6 and 4.9, respectively). For
these specific values it is just
as likely to be Virginica and
```

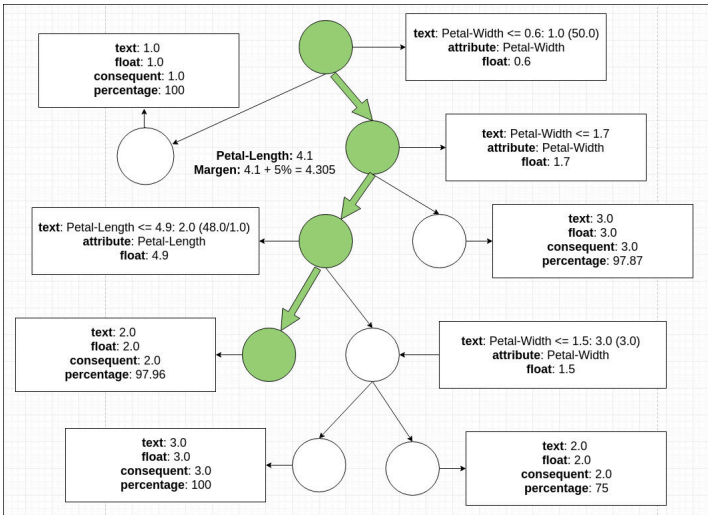


Figura 3. Clasificación de la instancia Sepal-Length: 5.6, Sepal-Width: 3, Petal-Length: 4.1, Petal-Width: 1.3.

```
Versicolor. But Virginica and
Versicolor will be an exception
because class Setosa is confused
with Virginica and Versicolor
only in 2% and 0% of cases,
respectively.
```

En este caso, la clasificación realizada es como clase Setosa. Sin embargo, al ser los valores de la instancia idénticos a los umbrales, y entrar en el rango de la tolerancia establecida del 5 %, se consideran como alternativas las dos ramas del nodo raíz y las del nodo que clasifica por longitud. Todas estas alternativas conducen a las clases Virginica y Versicolor. En ambos casos se indica el valor umbral que lo justifica y se valora la situación como que podría ser indistintamente tanto una como otra. Sin embargo, se introduce un matiz de carácter global, puesto que de acuerdo con la matriz de confusión del clasificador, la confusión de la clase Setosa con las clases Virginica y Versicolor es muy poco frecuente:

$$\begin{pmatrix} & Set. & Virg. & Vers. \\ Set. & 49 & 1 & 0 \\ Virg. & 0 & 47 & 3 \\ Vers. & 0 & 2 & 48 \end{pmatrix}$$

### *IV-B. Conjunto de datos FLAVIA*

En esta sección discutimos un caso más realista. FLAVIA[1] es un proyecto de código abierto en el que se abordó la creación de un conjunto de datos para la clasificación automática de hojas en la región de Yangtze Delta (próxima a Shanghai) en China. El conjunto de datos está formado por 1800 muestras de hojas (15 atributos) que corresponden a 32 clases diferentes. Una red neuronal es capaz de clasificar todas las hojas con una tasa de acierto superior al 90 % [19]. Sin embargo, la clasificación se basa en un modelo de caja negra que una persona no puede entender. El árbol construido por el algoritmo J48 de Weka contiene 449 nodos (225 nodos-hoja) y una tasa de acierto de clasificación del 70.44 % (considerando *10-fold cross-validation*). Se puede apreciar cómo pasar de un modelo de caja negra a un modelo de caja blanca supone en este caso una reducción apreciable en precisión. Además, aunque el modelo generado es de caja blanca, el elevado número de clases, atributos y nodos hace que la interpretación no sea sencilla, incluso para un experto en botánica.

En [20], presentamos los resultados de una encuesta en la que demostramos la utilidad de generar explicaciones en lenguaje natural asociadas a clasificaciones hechas por un conjunto de reglas borrosas aprendidas sobre un subconjunto de los datos de FLAVIA, con 310 instancias, 3 atributos y 5 clases (Fig. 4). De los 15 atributos de partida (que caracterizan propiedades geométricas y morfológicas) seleccionamos sólo los tres (Área, Perímetro y Diámetro) que un experto en botánica consideró útiles a fin de explicar en lenguaje natural el proceso de clasificación; prestando atención únicamente a la forma de la hoja. En esta sección, consideramos el mismo conjunto de datos usado en [20]. La explicación global es la siguiente:

```
There are 5 types of flavia:
Aesculus chinensis, Berberis
anhweiensis, Cercis chinensis,
Phoebe zhennan and Lagerstroemia
indica.
This classifier is very reliable
because correctly classified
instances represent 90.97%. There
may be some confusion among samples
related to some types of flavia.
But among all of them the pair
[Cercis chinensis; Phoebe zhennan]
is the most confused.
```

La matriz de confusión correspondiente (ordenada según las 5 clases de hoja en la Fig. 4) es:

$$\begin{pmatrix} 60 & 1 & 2 & 0 & 0 \\ 0 & 55 & 2 & 1 & 0 \\ 3 & 0 & 63 & 6 & 0 \\ 1 & 0 & 5 & 52 & 2 \\ 0 & 0 & 1 & 4 & 52 \end{pmatrix}$$

El árbol construido en este caso contiene sólo 25 nodos (13 nodos-hoja) y una tasa de acierto de clasificación del 90.97 % (considerando *10-fold cross-validation*).

ExpliClas permite introducir a mano el valor numérico de los atributos cuando el objeto a clasificar no coincide con ninguna de las instancias en el conjunto de datos. Por ejemplo, la explicación local para la hoja en la Fig. 5 sería la siguiente. Nótese que una sencilla comparativa visual entre las figuras 4 y 5 permite verificar cualitativamente la explicación dada.

```
Flavia is type Cercis chinensis
because its area is not very
small and its perimeter is small.
However, this flavia may be
```

[1] http://flavia.sourceforge.net/

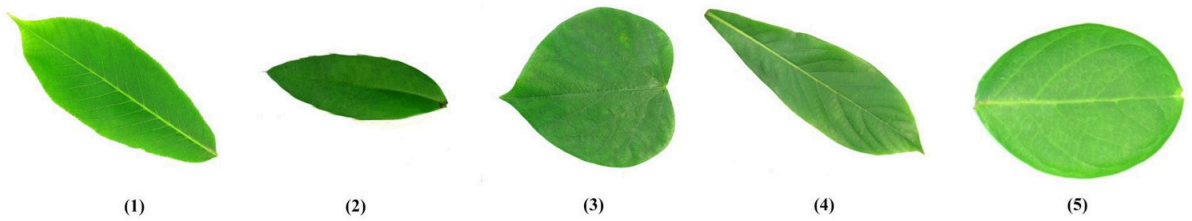


Figura 4. Las 5 clases en la versión reducicada de FLAVIA: (1) Aesculus chinensis, (2) Berberis anhweiensis, (3) Cercis chinensis, (4) Phoebe zhennan, (5) Lagerstroemia indica.

Figura 5. Ejemplo de hoja a clasificar (Área: 349.045, Perímetro: 2.964,304, Diámetro: 666,647).

```
also Phoebe zhennan because its
perimeter is quite close to the
split value (3,042.19). For these
specific values it is just as
likely to be Phoebe zhennan. But
Phoebe zhennan will be an exception
because class Cercis chinensis is
confused with Phoebe zhennan in
8.33% of cases.
```

## V. Conclusiones y Trabajo Futuro

En este trabajo hemos presentado un modelo para la generación de explicaciones (globales y locales) en lenguaje natural sobre clasificaciones hechas con árboles de decisión con atributos numéricos. El modelo está implementado en el servicio web ExpliClas [1]. Como trabajo futuro, realizaremos una validación exhaustiva del modelo con usuarios reales y refinaremos las explicaciones según la realimentación recibida. Adicionalmente, extenderemos el modelo de explicación para considerar atributos categóricos y algoritmos de clasificación de caja gris, como árboles de decisión borrosos, entre otros.

## Agradecimientos

Jose M. Alonso es Investigador Ramón y Cajal (RYC-2016-19802). Este trabajo está financiado por los proyectos TIN2017-90773-REDT (iGLN), TIN2017-84796-C2-1-R (BIGBISC), TIN2014-56633-C3-1-R (BAI4SOW) y TIN2014-56633-C3-3-R (ABS4SOW) (Ministerio de Economía y Competitividad) y GRC2014/030 y "Acreditación 2016-2019, ED431G/08" (Xunta de Galicia), todos con cofinanciación FEDER.

## Referencias

[1] B. López-Trigo, J. M. Alonso, and A. Bugarín, "ExpliClas: Web service for the automatic explanation in natural language of classification models in data mining," 2018, http://demos.citius.usc.es/ExpliClas/.

[2] K. Panetta, "Gartner top 10 strategic technology trends for 2018," 2017, https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/.

[3] S. Barocas and D. Boyd, "Computing ethics. engaging the ethics of data science in practice," *Communications of the ACM*, vol. 60, no. 11, pp. 23–25, 2017.

[4] Parliament and Council of the European Union, "General data protection regulation (GDPR)," 2016, http://data.europa.eu/eli/reg/2016/679/oj.

[5] D. Gunning, "Explainable Artificial Intelligence (XAI)," Defense Advanced Research Projects Agency (DARPA), Arlington, USA, Tech. Rep., 2016, DARPA-BAA-16-53.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? explaining the predictions of any classifier," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, San Francisco, USA, 2016, pp. 1–10.

[7] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, "Generating visual explanations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 3–19.

[8] K. Darlington, "Explainable AI systems: Understanding the decisions of the machines," 2017, openMind, BBVA Group, https://www.bbvaopenmind.com/en/explainable-ai-systems-understanding-the-decisions-of-the-machines/.

[9] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[10] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[11] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classication and Regression Trees*, 1st ed. Wadsworth, 1984.

[12] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, 2018.

[13] E. Reiter and R. Dale, *Building natural language generation systems*. Cambridge University Press, 2000.

[14] L. A. Zadeh, "A new direction in AI: Toward a computational theory of perceptions," *Artificial Intelligent Magazine*, vol. 22, no. 1, pp. 73–84, 2001.

[15] A. Gatt and E. Reiter, "SimpleNLG: a realisation engine for practical applications," in *Proceedings of the European Workshop on Natural Language Generation (ENLG)*, Athens, Greece, 2009, pp. 90–93.

[16] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Morgan Kaufmann, 2016.

[17] The University of Waikato, "Weka 3: Data Mining Software in Java," 2018, https://www.cs.waikato.ac.nz/ml/weka/.

[18] The University of California at Irvine, "UCI machine learning repository," 2018, https://archive.ics.uci.edu/ml.

[19] S. Gang Wu, F. Sheng Bao, E. You Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *IEEE International Symposium on Signal Processing and Information Technology*, 2007, pp. 1–6.

[20] J. M. Alonso, A. Ramos-Soto, E. Reiter, and K. van Deemter, "An exploratory study on the benefits of using natural language for explaining fuzzy rule-based systems," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Naples, Italy, 2017, pp. 1–6, http://dx.doi.org/10.1109/FUZZ-IEEE.2017.8015489.