



Escuela Politécnica
Superior (EPS)



UNIVERSIDAD DE CÓRDOBA
Departamento de
Informática y
Análisis Numérico

Práctica 2. Preprocesamiento y más filtros de Weka

Juan Carlos Fernández Caballero (jfcaballero@uco.es)
Introducción al Aprendizaje Automático (IAA)
3º de Grado de Ingeniería Informática
Especialidad en Computación
Curso 2019-2020



GRUPO DE INVESTIGACIÓN AYRNA
APRENDIZAJE Y REDES NEURONALES ARTIFICIALES
uco.es/ayrna

Agradecimientos

- Parte de estas diapositivas se han elaborado con la colaboración del grupo AYRNA de la Universidad de Córdoba (<https://www.uco.es/ayrna/>) y del Ingeniero Antonio Manuel Gómez Orellana (am.gomez@uco.es), como parte de su participación en el proyecto docente de Kaggle vinculado a esta asignatura en el curso 2018-2019.

Índice de contenido

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

¿Por qué preprocessar los datos?

Los datos del mundo real están “*sucios*” y pueden aportar ruido:

- **Incompletos:** Datos perdidos.
 - **Mala representación y consistencia en el formato:**
 - ▶ La codificación puede que no sea homogénea.
Ej: Edad en años/días.
Ej: Formato de números.
Ej: Fechas: 2020-03-04; 4/3/2020.
 - **Duplicidad:** Existencia de duplicidad de patrones.
 - **Mediciones erroneas:** Errores en la toma o transcripción de datos.
 - **Patrones irrelevantes:** Existencia de patrones que se deban eliminar dependiendo del valor que tengan en un determinado atributo.

¿Por qué preprocessar los datos?

Los datos del mundo real están “*sucios*” y pueden aportar **ruido**:

- **Valores atípicos y extremos:** Existencia de *outliers* y casos extremos.
 - **Redundancias en atributos:**
 - ▶ **¿Existen atributos redundantes?:** Análisis de correlaciones y selección de características.
 - ▶ **¿Existen atributos de diferentes fuentes que representen lo mismo?:** Distinto a correlación.
 - ▶ **¿Existen atributos que no aporten información?:** Identificadores.

Este tipo de datos **no son útiles** para los algoritmos de aprendizaje.

Objetivo del preprocesamiento

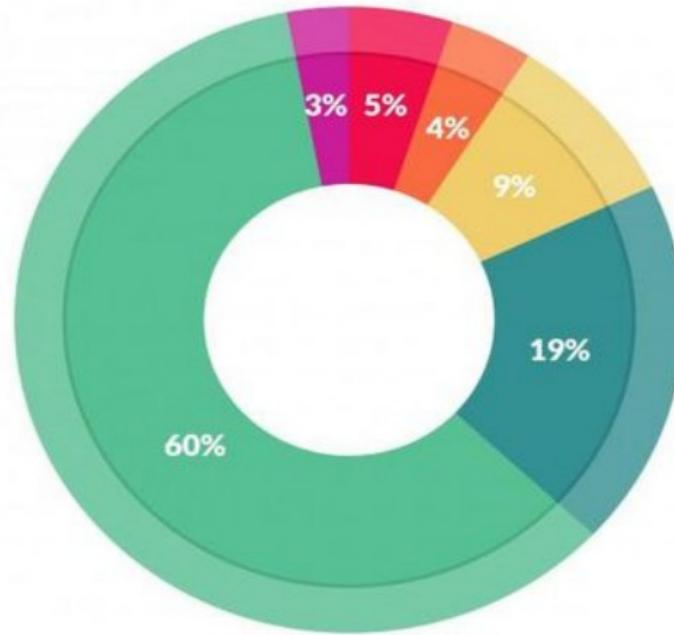
Mejorar la calidad de los datos, de forma que:

- Sean interpretables por los algoritmos.
 - Se pueda inferir (extraer) el máximo conocimiento.
 - Se consiga un mejor rendimiento.

Datos de calidad → Resultados de calidad.

- El preprocessamiento es uno de los **procesos más importantes en el flujo de acciones sobre un conjunto de datos**. Es determinante para la obtención de modelos con buen rendimiento.
 - Cada conjunto de datos necesita un **preprocessamiento concreto** y diferente del realizado a otros.
 - Está considerado como un problema desafiante en las tareas de investigación [1].

Proceso muy importante y laborioso



What data scientists spend the most time doing

- Building training sets: 3%
 - Cleaning and organizing data: 60%
 - Collecting data sets; 19%
 - Mining data for patterns: 9%
 - Refining algorithms: 4%
 - Other: 5%

Figura 1: El preprocessamiento representa el 60 % del trabajo del científico de datos [2].

Tareas

Algunas de las tareas involucradas en el **preprocesamiento** de datos (no necesariamente en este orden):

- Preparación: Formateo y consistencia.
 - Visualización.
 - Tratamiento de datos perdidos.
 - Transformación (normalización, binarización, etc).
 - Tratamiento de *outliers* y de valores extremos.
 - Selección de características.
 - Selección de instancias.

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

Preparación: Formateo y consistencia.

Algunos filtros de Weka para preparación de datos:

- ***filters/unsupervised/instances/RemoveDuplicates***: Elimina patrones duplicados.
 - ***filters/unsupervised/instances/RemoveWithValues***: Elimina patrones conforme al valor de un atributo.
 - ***filters/unsupervised/instances/RemoveRange***: Elimina patrones en función de su índice en el conjunto de datos.
 - ***filters/unsupervised/attribute/Remove***: Elimina atributos en función de su índice.
 - ***filters/unsupervised/attribute/RemoveType***: Elimina atributos en función de su tipo.
 - ***filters/unsupervised/attribute/Reorder***: Reordena atributos en función de su índice.
 - ***filters/unsupervised/attribute/SortLabels***: Reordena las etiquetas de los atributos nominales.
 - ***filters/supervised/attribute/ClassOrder***: Reordena las clases.
 - ***filters/supervised/attribute/MathExpression***: Modifica atributos numéricos en función de una expresión matemática proporcionada.

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

Visualización de los datos

Convertir la información en una **representación gráfica**, la cual ofrece una **visión más coherente** de los datos.

Posibles ventajas al visualizar datos:

- Obtención de una **comprensión más detallada** del problema y sus datos.
 - Detectar posibles datos erróneos.
 - Detectar y/o comparar posibles **tendencias** o patrones, **frecuencias inusuales**.
 - Ayuda a **enfocar las tareas de preprocesamiento** a realizar.

Técnicas de representación (I)

Histogramas:

- Visión de la distribución de la población respecto a una característica.
 - Muestran el **grado de homogeneidad** o de **variabilidad** de los datos.
 - Muestran frecuencias inusuales que podrían venir de un valor etiquetado incorrectamente.



Figura 2: Histograma atributo SalePrice.

Técnicas de representación (II)

Diagramas de caja: *boxplot*

- Proporcionan el valor máximo, el mínimo, la mediana y los cuartiles.
 - Ofrecen una visión de la simetría y dispersión que siguen los datos.
 - Desvelan la presencia de posibles *outliers* y valores extremos.
 - https://es.wikipedia.org/wiki/Diagrama_de_caja



Figura 3: Boxplot atributo SalePrice.

Técnicas de representación (III)

Gráficos de dispersión: *scatter plot*

- Estudian la relación existente entre dos atributos.
 - Pueden sugerir correlaciones entre los atributos.
 - Muy útiles para detectar *outliers* y valores extremos.
 - https://es.wikipedia.org/wiki/Diagrama_de_dispersi%C3%B3n

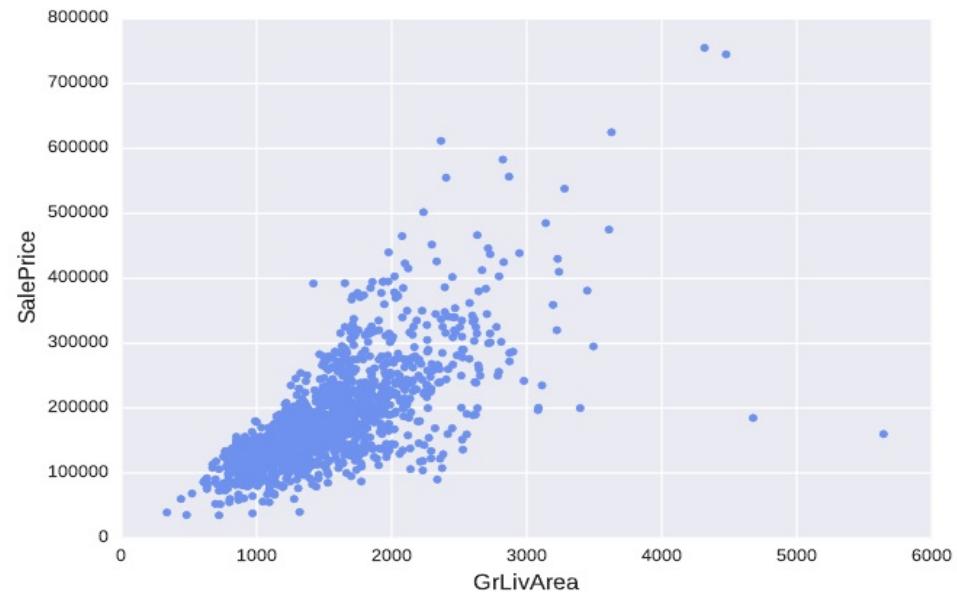


Figura 4: Scatter plot atributo SalePrice.

Visualización de los datos en Weka (I)

Preprocess

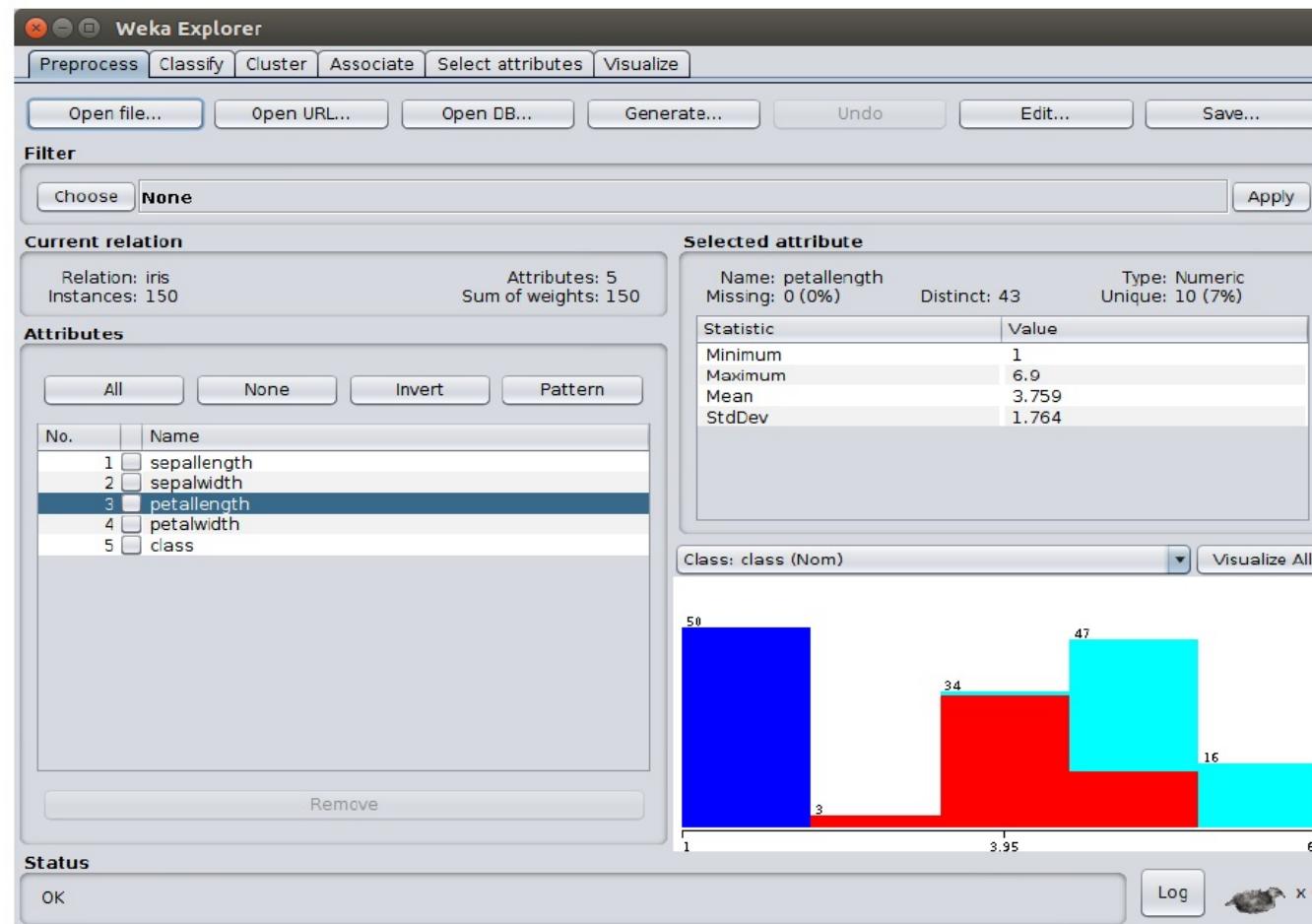


Figura 5: Histograma del atributo petallength

Visualización de los datos en Weka (II)

Visualize

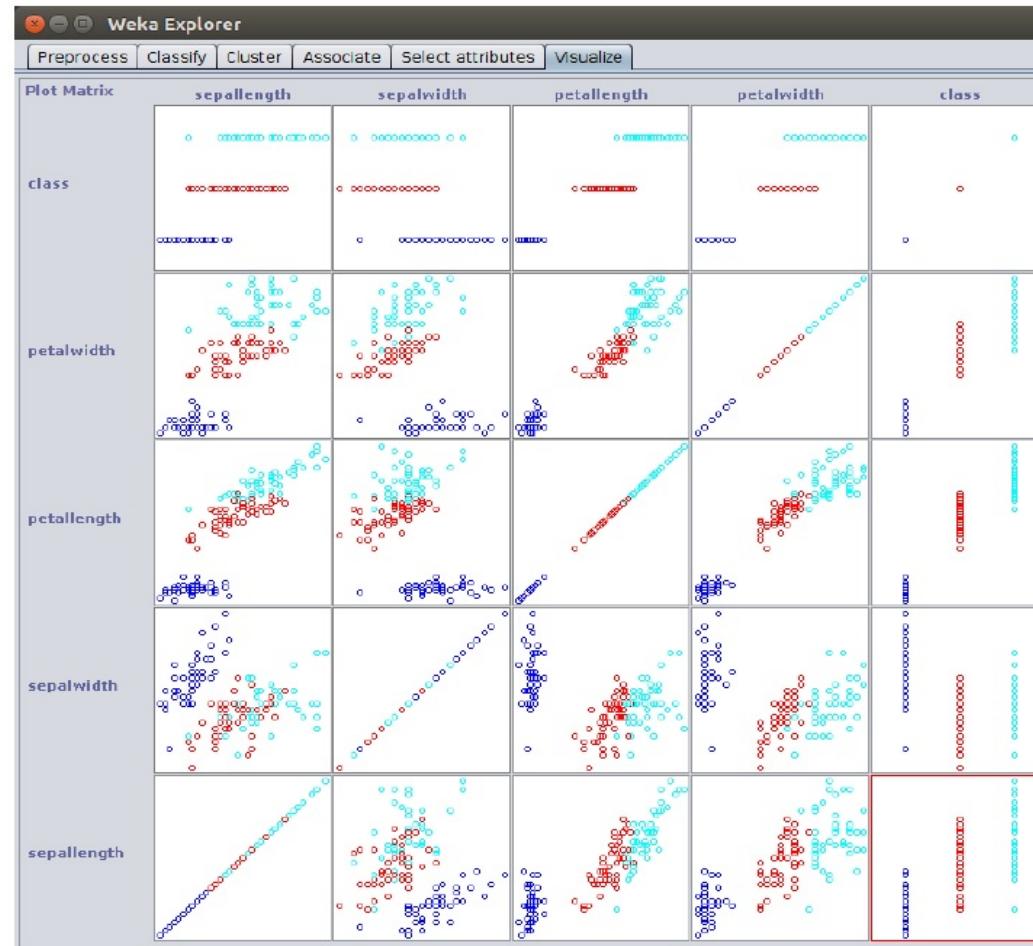


Figura 6: Scatter plots (dispersión) pares de atributos conjunto de datos Iris.

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

Datos perdidos

Una situación a la que se enfrenta frecuentemente cualquier científico de datos es el tratamiento de los valores perdidos.

- Los valores perdidos son aquellos que para una variable determinada **no constan en algunas filas o patrones**.

Ej: Fallos en los instrumentos de medida.

Ej: Sujetos que no asisten a la entrevista o no contestan a determinadas preguntas.

Problemas:

- Los datos perdidos producen **perdida de información**.
 - Introducen mucho **sesgo** (diferencia notable entre los datos observados y los no observados).
 - **Ajustes no deseados** de los modelos a los datos.

<https://conocemachinelearning.wordpress.com/tag/valores-perdidos>

Ejemplo

En Weka se indican con '?' o 'NaN'.

	name	gender	height	weight	age
0	Michael	None	123.0	10.0	14.0
1	Jessica	F	145.0	NaN	NaN
2	Sue	NaN	100.0	30.0	29.0
3	Jake	F	NaN	NaN	NaN
4	Amy	NaN	NaN	NaN	52.0
5	Tye	M	150.0	20.0	45.0

Figura 7: Conjunto de datos con datos perdidos.

Tratamiento de los datos perdidos

No es una regla exacta.

En atributos:

- Datos perdidos $\geq 40\%$, se podría eliminar el atributo.
 - En caso contrario, se imputan (recuperan) los datos perdidos.

En patrones:

- Datos perdidos $\geq 50\%$, se podría eliminar el patrón.
 - En caso contrario, se imputan los datos perdidos.

Eliminación de datos perdidos

Con la eliminación de datos **se pierde información** que puede ser muy **valiosa** para el modelo.

Eliminar atributos cuando el imputar datos perdidos:

- Aporte poca información (atributo no relevante).
 - Genere mucho “ruido” (información sintética poco real).

Eliminar patrones cuando:

- Dispongan de poca información interesante para el modelo (muy incompletos: muchos atributos = NaN).

Recuperación (imputación) de datos perdidos (I)

Cuando sea posible, es interesante **recuperar los datos perdidos**.

Existen muchas técnicas en el estado del arte:

- Reemplazar el valor a mano → es impracticable.
 - Reemplazar por la media (moda, mediana) del conjunto de datos.
 - Regresión entre otros atributos.
 - Mediante técnicas de *Machine Learning* → *kNN*, *clustering*, etc.

Recuperación (imputación) de datos perdidos (II)

Reemplazar por la media del conjunto de datos

Es una técnica cómoda y sencilla. También se puede utilizar la mediana o la moda, según el tipo de atributo. **Es un poco más justa cuando se emplean patrones de la misma clase.**

Regresión entre atributos

Se establece una **regresión entre atributos (sin datos perdidos)** y así poder imputar los valores que faltan de los demás.

Mediante técnicas de *Machine Learning*

kNN: Se pueden imputar los datos perdidos de un patrón en base a sus K muestras más próximas en el espacio de atributos (distancia euclídea por ejemplo), y sustituir por la media o moda de esos K más cercanos [3].

Recuperación (imputación) de datos perdidos (III)

Ejemplo tomando la **media**:

Person	Highest Education	Salary
A	School	10000
B	Post Graduate	40000
C	Graduate	35000
D	School	11000
E	Graduate	NA
F	Post Graduate	42000
G	Post Graduate	39000
H	Graduate	25000
I	School	12000
J	School	NA
K	Graduate	31000
L	Post Graduate	39500

- Media global: 28450 (mismo salario para *School* que para *Graduate*).
 - **Solución:** Calcular la media por separado para *School* (11000) y para *Graduate* (30333).

Recuperación (imputación) de datos perdidos (IV)

Ejemplo tomando la **moda**:

Product	Type	User Rating (0-5)
A	Grocery	4.5
B	Cream	4.0
C	Fashion	4.5
D	Fashion	4.0
E	Cream	NA
F	Fashion	4.5
G	Grocery	4.0
H	Cream	4.5
I	Cream	4.0
J	Grocery	4.5
K	Fashion	4.0
L	Grocery	4.5

- En este caso, no podemos usar medias, ya que la variable es categórica ordinal.
 - El valor medio sería 3,18 y no tiene sentido como categoría.
 - Deberíamos calcular la moda (preferentemente condicionada a *Cream*).

Recuperación (imputación) de datos perdidos en Weka

Filtros de Weka **no supervisados** a nivel de atributo:

- ***filters/unsupervised/attribute/ReplaceMissingValues***: Reemplaza los datos perdidos de cada atributo por su media.
 - ***filters/unsupervised/attribute/ReplaceMissingWithUserConstant***: Reemplaza los datos perdidos de cada atributo por el valor suministrado por el usuario.
 - ***filters/unsupervised/attribute/AddValues***: Añade una etiqueta a valores perdidos.

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

Transformación de los datos

Normalización min-max: Transformación lineal de los datos, normalmente entre [0,1], de forma que **todos los atributos dominen por igual** (misma importancia).

Los nuevos datos están en el mismo rango y conservan la relación entre los datos originales.

Normalización: Weka

filters/unsupervised/attribute/Normalize: Normaliza los atributos numéricos del conjunto de datos.

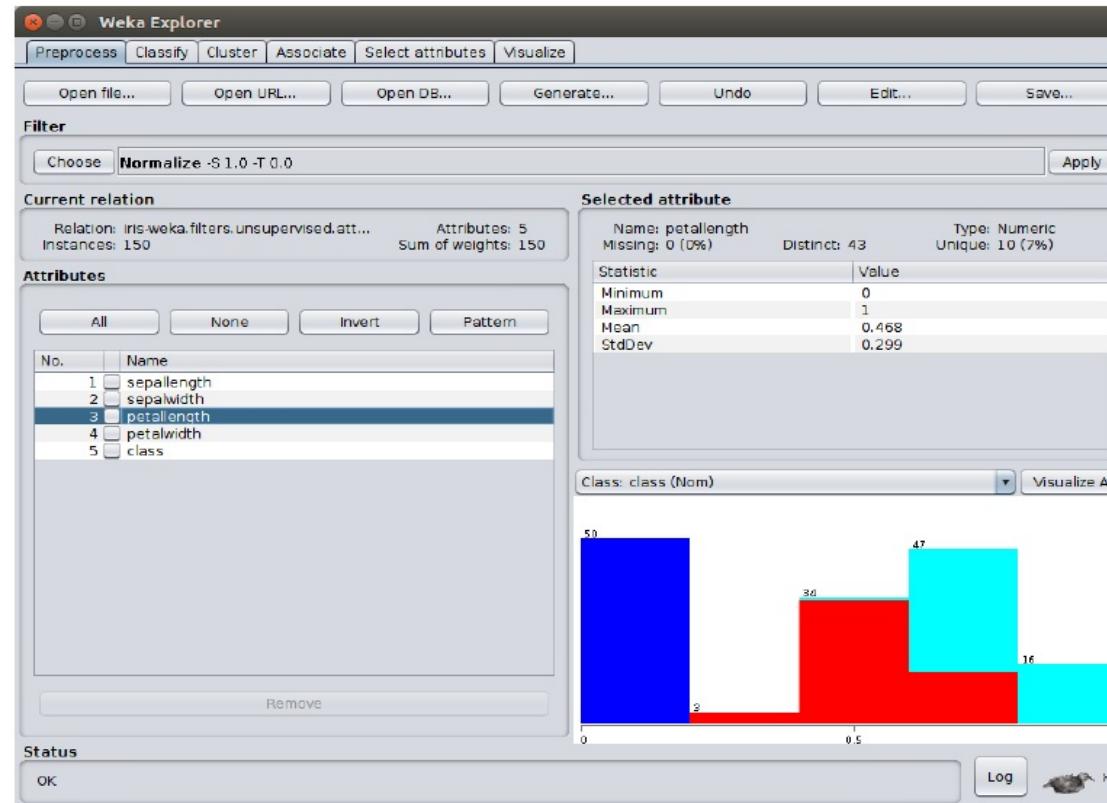


Figura 8: Normalización del conjunto de datos Iris.

Discretización

Algunos algoritmos **trabajan solo con atributos nominales**, o en ocasiones hay **necesidad de discretizar** una variable.

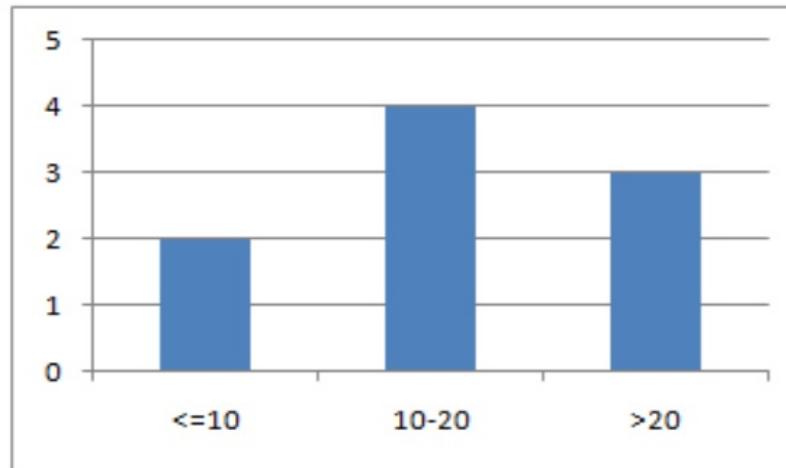
- Por ejemplo, dispongo de una variable edad que toma valores de 5 a 64 años.
 - Genero una variable categórica nominal con estas categorías:
`{edad<=10, 10<edad<=30, 30<=edad<45, edad>=45}`
 - Representar una variable con valores discretos permite **reducir la cantidad de información** y hacer que los atributos sean **más fáciles de entender**.
 - Algoritmos de discretización **no supervisados**:
 - ▶ Igual amplitud.
 - ▶ Igual frecuencia.
 - ▶ *Clustering* (*k-medias...*) Se basa en agrupar instancias similares.

Discretización

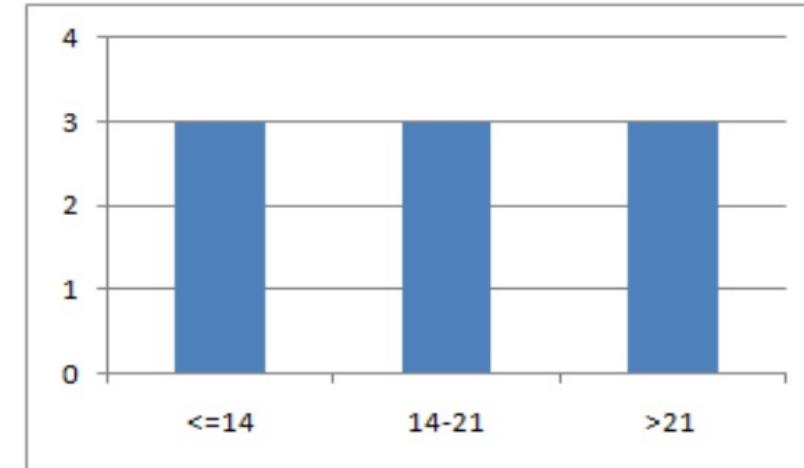
- Igual amplitud.

- ▶ Divide el intervalo en k intervalos del mismo ancho.
 - ▶ Si m es el valor mínimo y M es el valor máximo, el ancho será $W = \frac{M-m}{k}$.
 - ▶ Es la forma más simple, pero los *outliers* pueden dominar la conversión.
 - ▶ Además, puede generar desbalanceo de las categorías generadas.

Equal width



Equal frequency

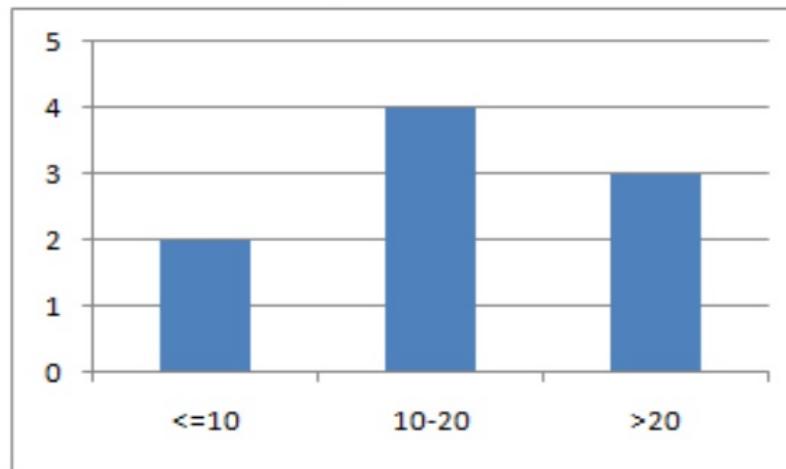


Discretización

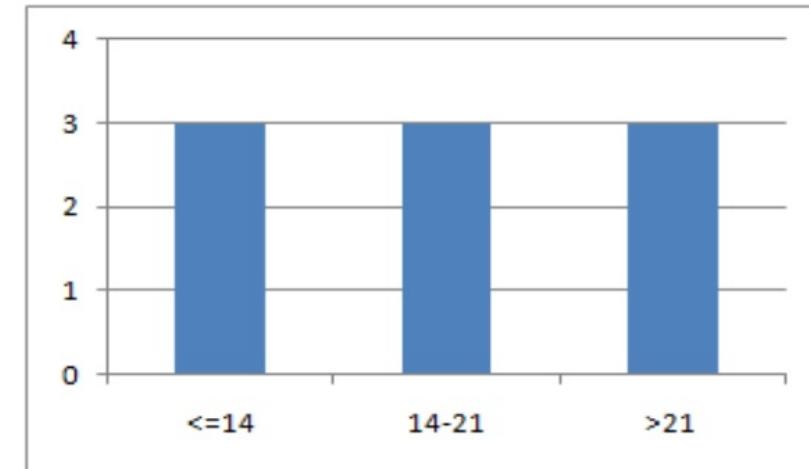
- Igual frecuencia.

- ▶ Divide el intervalo en k intervalos de distinto ancho, tratando de generar categorías balanceadas.
 - ▶ Es decir, se fuerza a que, tras la discretización, el número de ejemplos en cada categoría sea, aproximadamente, el mismo.

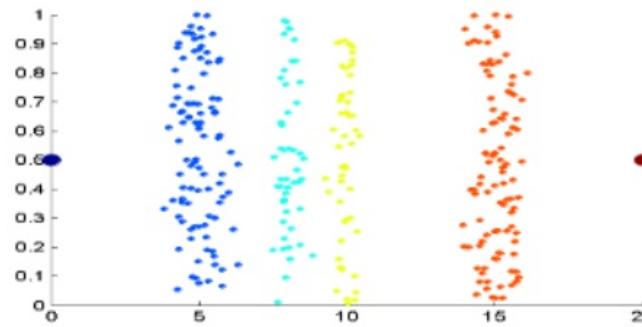
Equal width



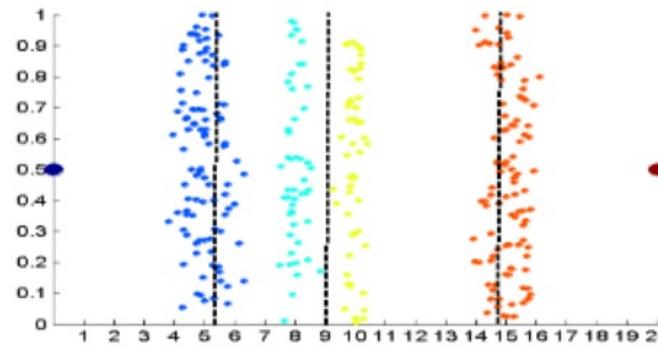
Equal frequency



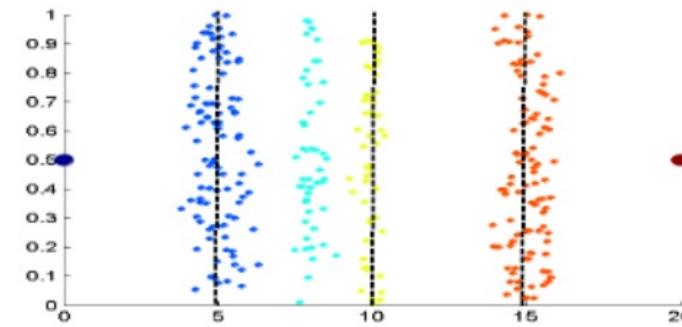
Discretización



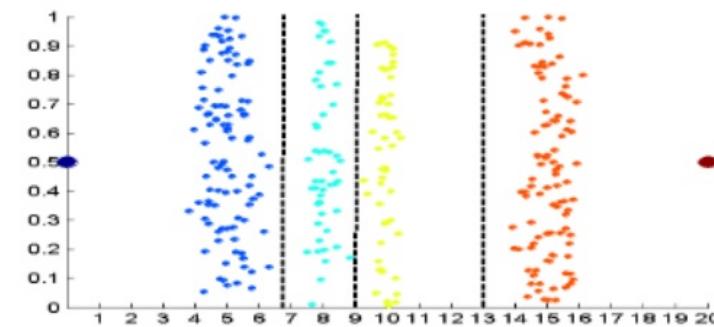
Datos



Igual frecuencia



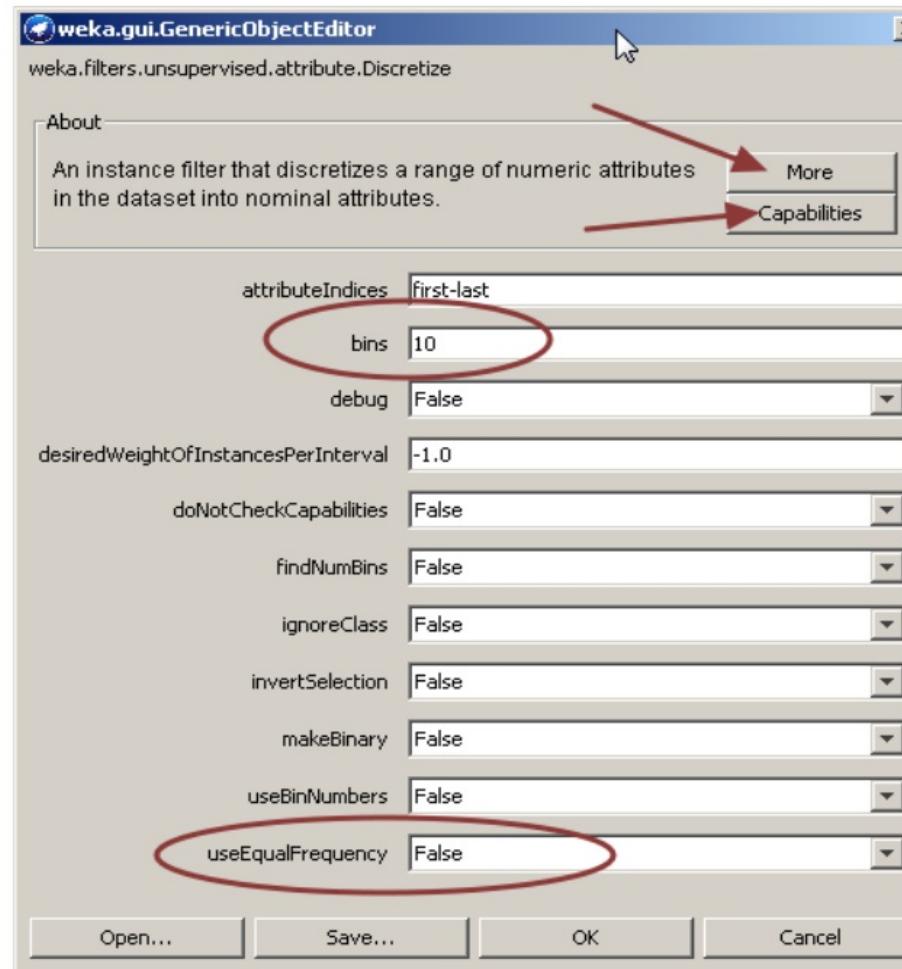
Igual anchura de intervalo



K-medias

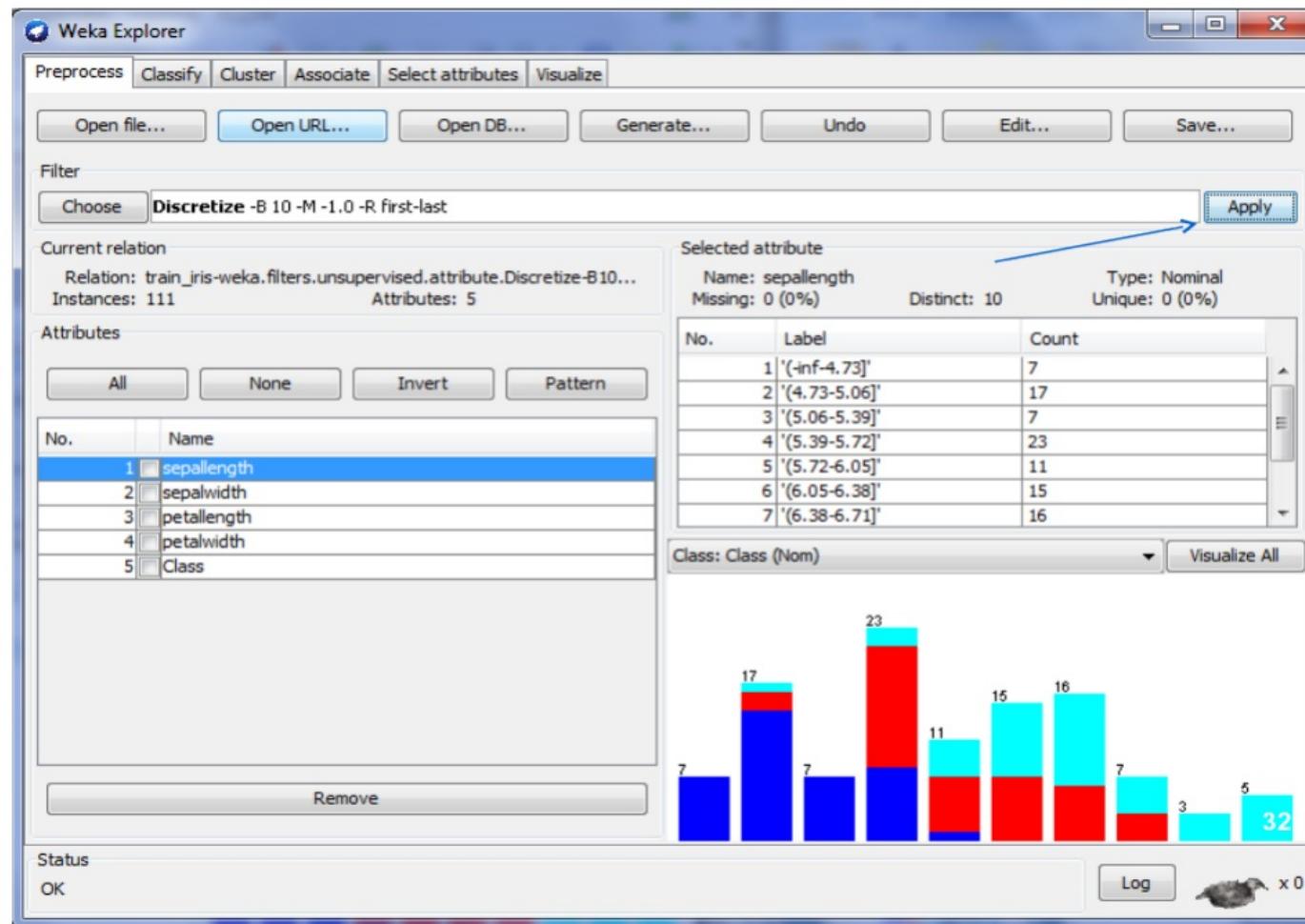
Discretización en Weka

filters/unsupervised/attribute/Discretize: Discretiza atributos por amplitud y frecuencia.



Discretización en Weka

Base de datos Iris discretizada en Weka por amplitud.



Nominales a binarios

Algunos métodos como las Redes Neuronales, la regresión o incluso algunos métodos de detección de *outliers* **solo trabajan con atributos numéricos**. Es necesaria una **transformación a valores numéricos**:

- En Weka filters→supervised→attribute→ NominalToBinary
 - Con la opción **BinaryAttributesNominal=False** (por defecto en Weka) todos los atributos nominales se transforman a numéricos (un nuevo atributo por cada etiqueta).
 - ▶ Los nominales que tenían **solo dos etiquetas en su lista** (<{etiqueta1, etiqueta2}) se transforman a numéricos con dos valores posibles, 0 ó 1.
 - ▶ Con la opción **transformAllValues=True** los nominales que tenían solo dos etiquetas en su lista también se transforman a numéricos (al igual que el resto de nominales) dando lugar a dos nuevos atributos binarizados.

Nominales a binarios

- Con la opción **BinaryAttributesNominal=True** los atributos nominales **siguen siendo nominales**, pero por cada etiqueta de la lista de nominales se crea un nuevo atributo nominal con dos etiquetas posibles {t,f}.
 - ▶ Los nominales que tenían **solo dos etiquetas en su lista** ({etiqueta1, etiqueta2}) permanecen igual.
 - ▶ Con la opción **transformAllValues=True** los nominales que tenían solo dos etiquetas en su lista también **siguen siendo nominales** (al igual que el resto de nominales) dando lugar a dos nuevos atributos nominales con dos etiquetas posibles {t,f}.

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

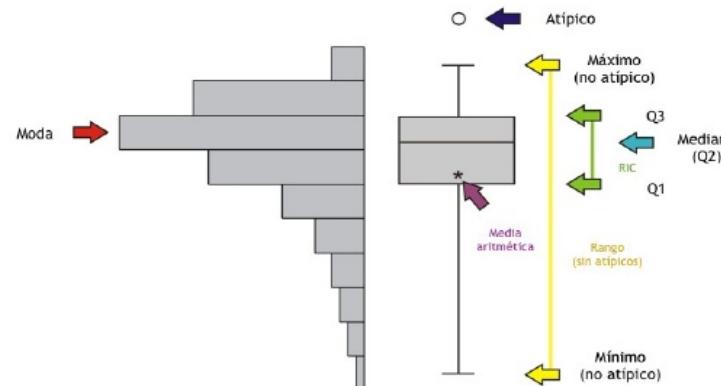
Entregables

Bibliografía

¿Qué es un *outlier*?

Es un patrón atípico comparado con el resto, es decir, tiene **valores de características considerablemente diferentes a la mayoría**.

- Influyen mucho sobre la media.
 - Detección: Mediante distancias (boxplots), mediante agrupamiento de patrones (*clustering*), otros métodos de *Machine Learning*...
 - Algunos libros refieren un valor como un *outlier* si este es mayor que 1.5 veces el valor del rango intercuartil ICR (diferencia entre el tercer y el primer cuartil) más alla de los cuartiles (gráficas *bloxplot*).
 - Pero **ojo**, un outlier **podría ser correcto** aunque sea anómalo estadísticamente (necesidad de un experto).



Ejemplo

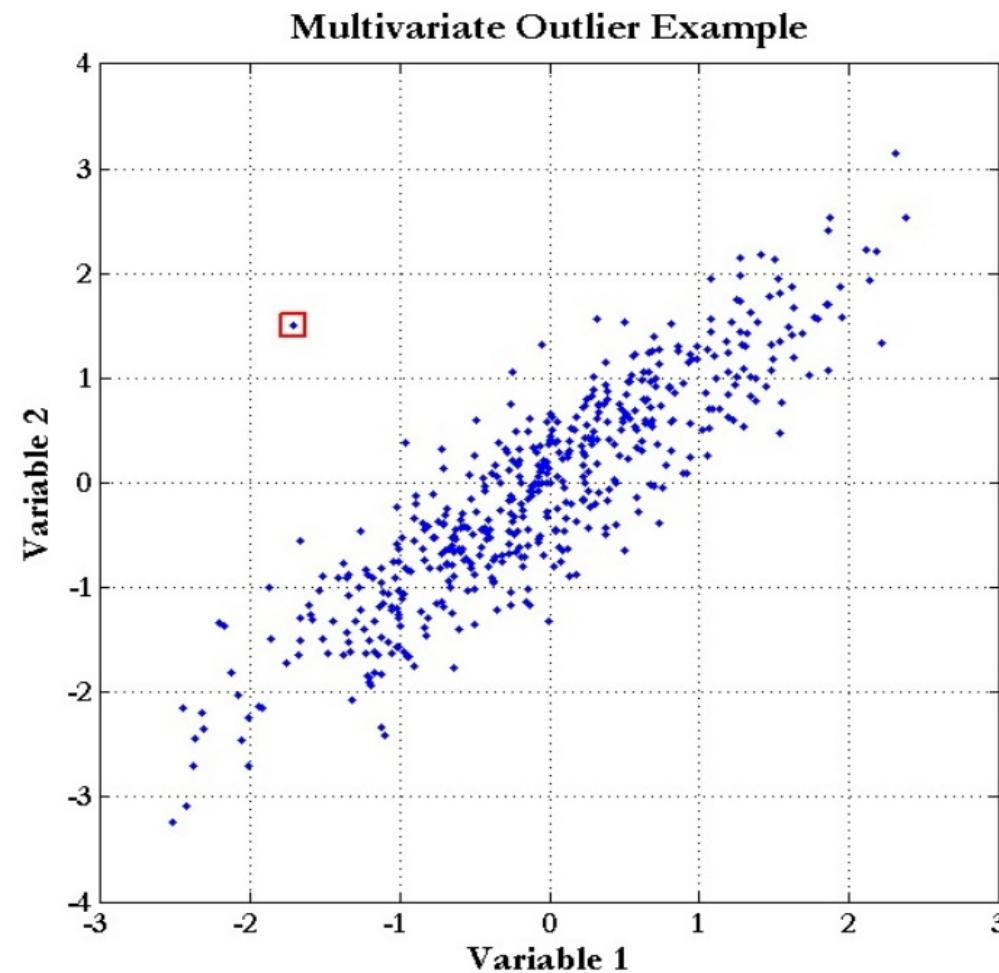


Figura 10: Detección de valores atípicos (outliers) mediante diagrama de dispersión.

Valores extremos

Un **valor extremo** o **atípico extremo** se podrá decir que es también un valor atípico o *outlier*, en el sentido de que un patrón puede tener unas características mucho más severas diferentes a los demás.

- La diferencia entre **outliers** o **atípicos** y **valores extremos** o **atípicos extremos** se puede definir por ejemplo mediante el valor del IRC.
 - Algunos autores difieren el valor a partir del cual un dato es atípico o atípico extremo.
 - ▶ valor $<$ (Q1-1.5IRC) o valor $>$ (Q3+1.5IRC) \longrightarrow atípico.
 - ▶ valor $<$ (Q1-2IRC) o valor $>$ (Q3+2IRC) \longrightarrow atípico extremo.
 - ▶ valor $<$ (Q1-3IRC) o valor $>$ (Q3+3IRC) \longrightarrow atípico extremo.
 - ▶ ...

Tratamiento de los *outliers* y los extremos

Es necesario detectar los *outliers* y extremos y dependiendo de la situación:

- **Ignorar**: Hay modelos que son robustos a *outliers* y extremos.
 - **Eliminar** el patrón.
 - **Reemplazar** el *outlier* o extremo por la media del atributo u otro estadístico.

Detección de *outliers* y extremos en Weka (I)

filters/unsupervised/attribute/InterquartileRange: Detecta atípicos y atípicos extremos. En cada patrón añade dos atributos adicionales que indican si éste se trata de un *outlier* o de un valor extremo. (Los atributos deben ser numéricos)

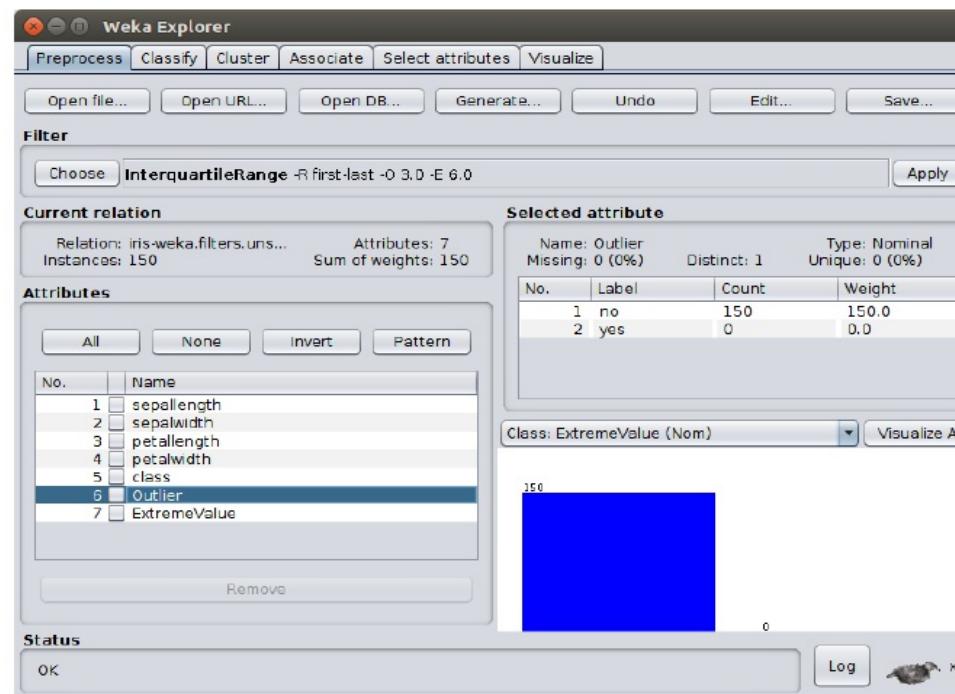


Figura 11: Detección de outliers en el conjunto de datos Iris.

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

Selección de características (SC)

Consiste en obtener una **representación reducida del conjunto de datos que preserve la información relevante** contenida en los datos originales.

- También conocida como selección de variables o de atributos o selección vertical [4].

Objetivos:

- Eliminar atributos que sean irrelevantes o redundantes, reduciendo la complejidad del problema.
 - Aumentar el rendimiento de los modelos.
 - Acelerar el proceso de aprendizaje.
 - Reducción del sobreajuste.
 - Proporcionar una mejor comprensión-interpretación del proceso subyacente que generó los datos, obteniendo modelos más reducidos.

SC por análisis de correlaciones

Una **correlación** indica la fuerza y dirección de una relación lineal entre dos variables:

- **Positiva:** Ambas variables cambian en la misma dirección.
 - **Neutra:** No hay relación en el cambio de las variables.
 - **Negativa:** Las variables cambian en direcciones opuestas.

El **rendimiento** de algunos algoritmos puede **deteriorarse** si **dos o más variables están estrechamente relacionadas**: **multicolinealidad**.

Eliminar una o varias de las variables correlacionadas puede **mejorar la precisión del modelo.**

SC por análisis de correlaciones

El coeficiente de correlación de *Pearson* devuelve un valor entre -1 y 1:

- **-1**: Correlación negativa completa.
 - **1**: Correlación positiva completa.
 - **0**: No hay correlación.

Valores $<(-0.6)$ o valores $>(0.6)$: Indica **correlación notable**.

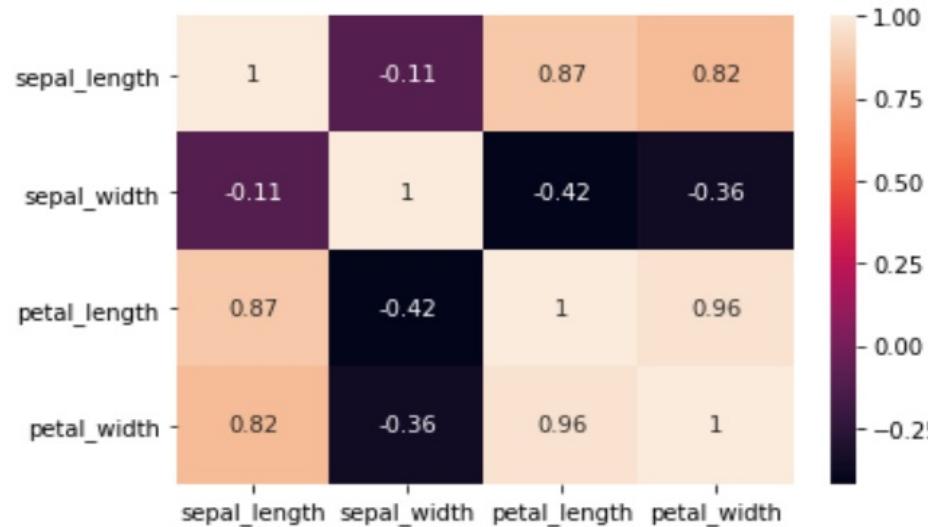


Figura 12: Matriz de correlación (mediante mapa de calor) del conjunto de datos Iris.

SC por análisis de correlaciones en Weka (I)

Para obtener desde Weka la **matriz de correlaciones** entre las variables de entrada seleccionar:

Select attributes → PrincipalComponents, Ranker

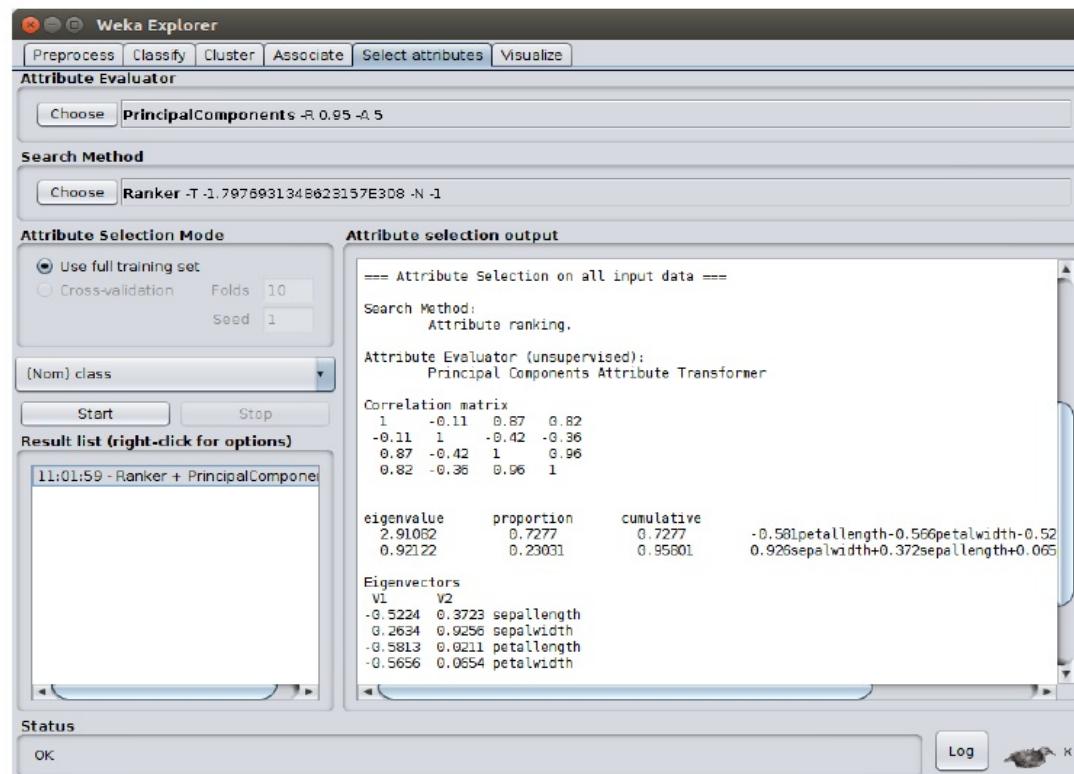


Figura 13: Matriz de correlación para el conjunto de datos Iris.

SC por análisis de correlaciones en Weka (II)

Para obtener desde Weka una valoración de cuánto **influye cada atributo (variable independiente)** sobre la predicción de salida:

Select attributes → CorrelationAttributeEval, Ranker

- **CorrelationAttributeEval** + **Ranker** ordena de mejor a peor los atributos respecto a la salida. Podríamos elegir los de mayor ranking y junto con la matriz de correlaciones hacer pruebas para ir comprobando rendimientos.
 - Este método se basa en el **coeficiente de correlación de Pearson** para obtener conclusiones de una variable independiente sobre la dependiente de salida.
 - Se suele utilizar en problemas de **regresión** para correlaciones lineales.
 - Si tenemos variables nominales se aconsejan pasarlas a numéricas (filtro **no supervisado NominalToBinary**).
 - Ojo, no indica por si solo la correlación entre los atributos independientes. Para ello es necesario obtener la matriz de correlaciones (**PrincipalComponents, Ranker**).

SC por análisis de correlaciones en Weka (III)

Otro método para obtener desde Weka una valoración de cuánto **influye cada atributo sobre la predicción de salida**, seleccionar:

Select attributes → InfoGainAttributeEval, Ranker

- **InfoGainAttributeEval + Ranker** ordena de mejor a peor los atributos respecto a la salida. Podríamos elegir los de mayor ranking y junto con la matriz de correlaciones hacer pruebas para ir comprobando rendimientos.
- Este método se basa en la **ganancia de información** para obtener conclusiones de una variable independiente sobre la dependiente de salida.
- Se suele utilizar en problemas de **clasificación**.
- **Ojo, no indica por si solo la correlación entre los atributos independientes.** Para ello es necesario obtener la matriz de correlaciones (**PrincipalComponents, Ranker**).

SC por análisis de correlaciones en Weka (IV)

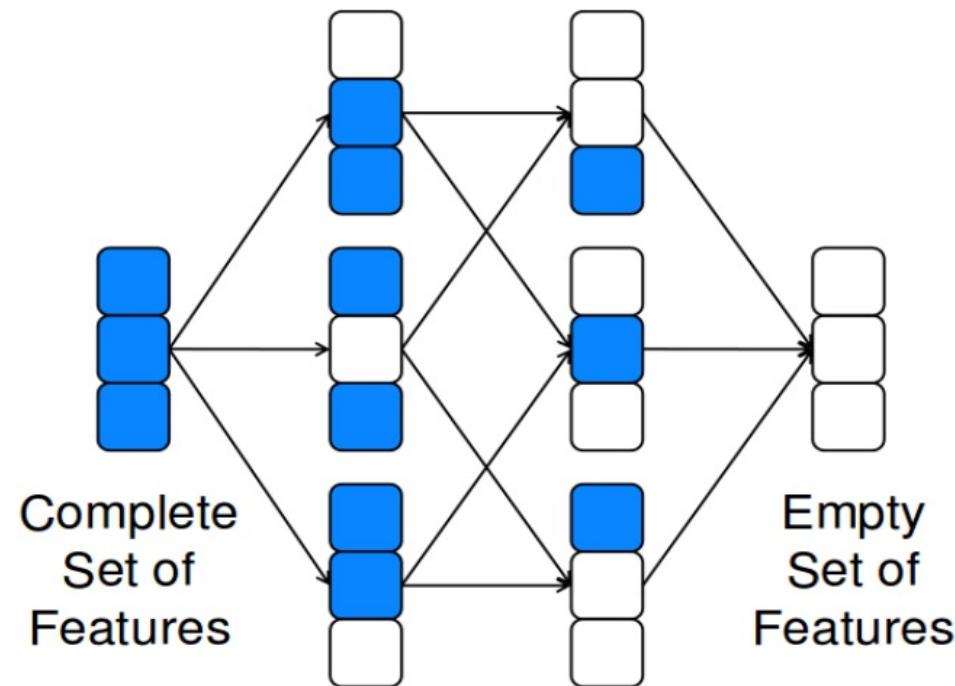
En **Moodle** dispone de dos ficheros .pdf donde se explica cómo hacer un análisis de correlaciones y la consiguiente eliminación de atributos basado en dicho análisis, para la base de datos *Iris*.

- “Obtención matriz de correlación en Weka para Iris”.
 - “Filtro *CorrelationAttributeEval* de Weka para Iris”.

Según ese ejemplo podríamos probar a eliminar el atributo “0.592 4 *petalwidth*”, con ***Remove*** en ***Preprocess***, y lanzar algún algoritmo de clasificación antes y después de eliminar dicho atributo, como por ejemplo en la pestaña ***Classify weka.classifiers.functions.logistic*** o ***weka.classifiers.functions.multilayerperceptron***.

SC mediante búsqueda + evaluación

- Otra manera de seleccionar característica es mediante **técnicas de búsqueda que exploren el espacio de los posibles subconjuntos de características**.
 - Se plantea como un problema de **búsqueda** y de **evaluación**.



SC mediante búsqueda + evaluación en Weka (I)

Método de evaluación: Mediante una función de evaluación se determina la **bondad** de subconjuntos de atributos en su **discriminación sobre la clase de salida**.

- Pestaña **Select attributes.Attribute Evaluator** en Weka (nos centraremos en dos).
 - ▶ **WrapperSubsetEval**: Evalúa un subconjunto de atributos mediante un **algoritmo de aprendizaje** (a configurar en Weka haciendo clic, por defecto **ZeroR**).Costoso porque necesita un proceso completo de entrenamiento y test por cada paso de búsqueda.
 - ▶ **CfsSubsetEval**: Calcula la correlación de la clase con cada atributo junto con un grado de redundancia respecto a los demás, eliminando atributos que tienen una correlación muy alta y atributos redundantes respecto a la salida.

SC mediante búsqueda + evaluación en Weka (II)

Método de búsqueda: Mediante una metodología de búsqueda se determinan la selección de subconjuntos de atributos.

- Determinados métodos pueden provocar problemas combinatorios inabordables cuando crece el número de atributos.
 - Otros métodos usan **estrategia (heurística)** para evitarlo. Es menos preciso pero también menos costoso.
 - Pestaña **Select attributes. Search Method** en Weka (nos centraremos en dos).
 - ▶ **Ranker**: No hace búsqueda de subconjuntos de atributos, sino que los ordena de mejor a peor según el algoritmo evaluador seleccionado. Seleccionar los k mejores.
 - ▶ **BestFirst**: Método de búsqueda voraz de subconjuntos de atributos.

SC mediante búsqueda + evaluación en Weka (III)

Combinación en la que nos centraremos en la práctica:

- ***CfsSubsetEval*** + ***BestFirst***: Selecciona un subconjunto de atributos del total que podrían ser representativos de nuestro problema. Debemos comprobar si obtenemos un rendimiento similar o mejor que con el conjunto total.

SC mediante búsqueda + evaluación en Weka (IV)

Select attributes → *CfsSubsetEval*, *BestFirst*

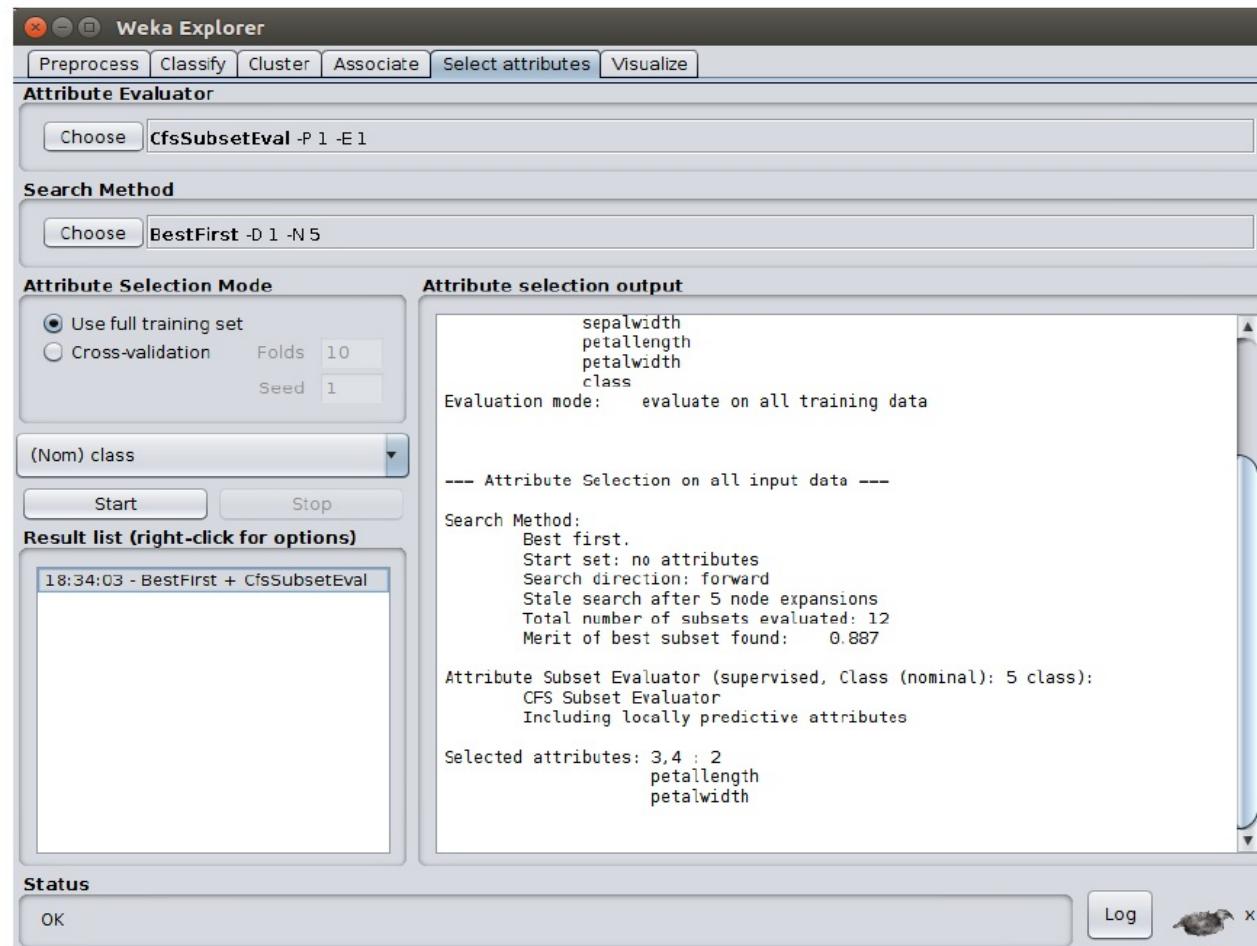


Figura 14: Selección de características *CfsSubsetEval + BestFirst* en conjunto de datos Iris.

Introducción

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

Selección de instancias

Consiste en obtener una **representación reducida del conjunto de datos que preserve la información relevante** contenida en los datos originales.

También conocida como selección de patrones o de ejemplos o selección horizontal.

Ventajas:

- Acelera el proceso de entrenamiento.
 - Mejor exactitud del modelo.
 - Modelos más simples e interpretables.
 - Reducción del ruido y patrones redundantes.
 - Facilita el aprendizaje con grandes volúmenes de datos.

Ejemplo

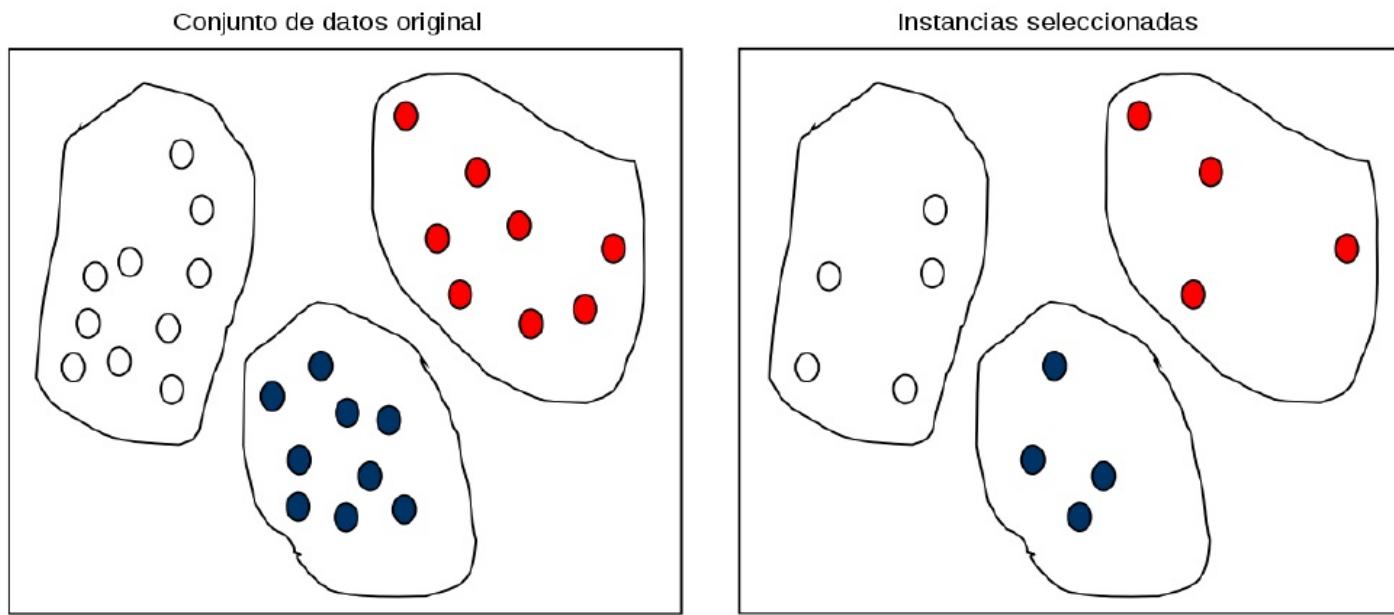


Figura 15: Ejemplo de selección de instancias.

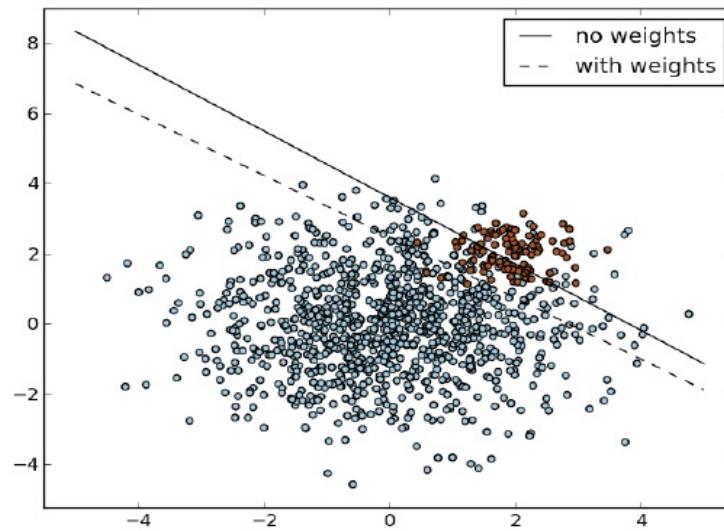
También se emplea para:

- Reducir el número de patrones (clases más numerosas).

Desbalanceo de los datos

En algunos casos las clases pueden tener una frecuencia extremadamente desigual:

- Diagnosis médica: 90 % saludables, 10 % enfermos.
 - Spam, fallos en maquinaria, etc.
 - ¡Mirar porcentaje de clasificación por clase!



Cuidado...

- Clasificador trivial: Aquel con un [90 %-99.99 %] correcto pero inútil.
 - Ej: 100 patrones (90 clase A y 10 clase B).
Reconoce a todos los de la clase A, pero ninguno de la clase B.

Desbalanceo de los datos

¿Qué hacer?

- **Sobremuestreo** (*over-sampling*): Generar nuevos patrones de la/s clase/s minoritaria/s (sintéticos, replicación, pesos, etc).
 - **Inframuestreo** (*under-sampling*): Seleccionar o eliminar una muestra de patrones de la/s clase/s mayoritaria.
 - Cuidado con la división train/test que se haya hecho → Usar **particiones estratificadas** por clase.

Selección de patrones y desbalanceo en Weka

Algunos filtros disponibles en Weka:

- ***filters/supervised/instance/SpreadSubsample***: Eliminación estratificada de patrones para cambiar las proporciones de las distintas clases del conjunto de datos original.
 - ▶ Parámetro *distributionSpread* = 1.0 nivela a la clase más pequeña.
 - ***filters/unsupervised/attributes/RemoveFolds***: Elimina (sin estratificar) subconjuntos aleatorios de patrones usando un *k-fold*.
 - ***filters/unsupervised/attributes/RemoveUseless***: Elimina atributos inútiles en función de un porcentaje de variación del total de los valores de los atributo.
 - ***filters/supervised/instance/Resample***: Cambia estratificadamente la proporción de patrones de las distintas clases del conjunto de datos original, con o sin reemplazo.
 - ***filters/unsupervised/instance/Resample***: Igual pero no usa estratificación.
 - ***filters/supervised/instance/ClassBalancer***: Cambia la proporción de patrones estratificadamente asignando unos pesos a los existentes (no añade ni elimina).
 - ***filters/unsupervised/instance/RemovePercentage***: Elimina un porcentaje de patrones de manera no estratificada.

Preparación

Visualización

Datos perdidos

Transformación

Outliers

Sel. características

Sel. instancias y desbalanceo

Entregables

Bibliografía

Entregables

Para esta práctica utilice el conjunto de datos de altura de ola proporcionado en Moodle.

1. Describa las operaciones de preprocessamiento que ha realizado sobre la base de datos proporcionada y cómo queda la base de datos final ya preprocessada. Se deja a su elección el conjunto de técnicas a aplicar, así como el nivel de detalle y descripción que quiera dar a su trabajo.

Para probar rendimientos sobre su preprocesamiento, puede lanzar cualquier algoritmo de Weka, por ejemplo

classifiers.functions.Logistic, y fijarse en la métrica “*Correctly Classified Instances*”. En la opción “*Supplied test set*” se indicaría el fichero del conjunto de test, mientras que el de entrenamiento corresponde al que se ha cargado desde la pestaña *Preprocess*.

Bibliografía adicional a la de la asignatura y al material de Moodle

- Q. Yang, X. Wu. 10 Challenging problems in data mining research. International Journal of Information Technology and Decision Making 5:4, 597-604., 2006.
 - G. Press, Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says, 2016.
 - Y. Obadia. The use of KNN for missing values, 2018.
 - H. Liu, H. Motoda. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic., 1998.
 - Weka 3: Data Mining Software in Java, 2019.
<https://www.cs.waikato.ac.nz/ml/weka>.
 - F. Herrera. Tema 5. Preparación de datos. Asignatura Inteligencia de Negocio., 2018.

¿Preguntas?