



Escuela Politécnica
Superior (EPS)



UNIVERSIDAD DE CORDOBA
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

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

Entregables

Bibliografía



Introducción

Preparación

Visualización

Datos perdidos

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

Entregables

Bibliografía

¿Por qué preprocesar 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 erróneas:** 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 preprocesar 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.
 - ▶ ¿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.

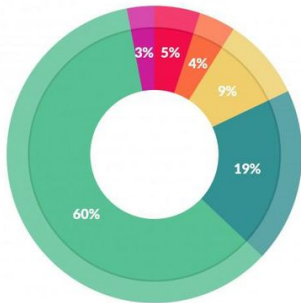
Objetivo del preprocesamiento

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.**

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 preprocesamiento representa el 60 % del trabajo del científico de datos [1].

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.
- Tratamiento de *outliers* y de valores extremos.
- Transformación (normalización, binarización, etc).
- Selección de características.
- Selección de instancias.

Introducción

Preparación

Visualización

Datos perdidos

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

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

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

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 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.



Figura 2: Histograma atributo SalePrice.

Gráficos de densidad:

- El concepto de frecuencia relativa se cambia por el de probabilidad.
- No dependen de los intervalos o número de “barras” usados en los histogramas.
- Muestran la **distribución de los datos de manera más precisa**. Es una línea continua que representa la distribución de densidad de toda la población.



Figura 3: Gráfico densidad 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

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: Boxplot atributo SalePrice.

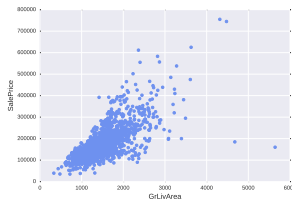


Figura 5: Scatter plot atributo SalePrice.

Visualización de los datos en Weka (I)

Preprocess

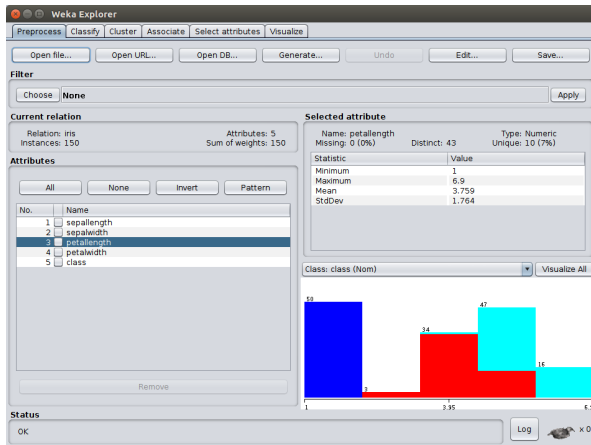


Figura 6: Histograma del atributo petallength.

Visualización de los datos en Weka (II)

Visualize

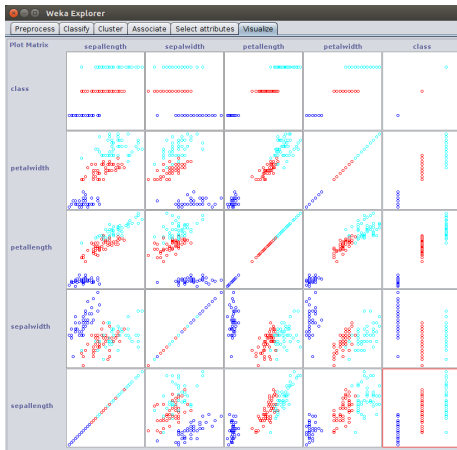


Figura 7: Scatter plots (dispersión) pares de atributos conjunto de datos Iris.

Introducción

Preparación

Visualización

Datos perdidos

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

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 8: 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 [2].

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

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

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 allá de los cuartiles (gráficas *bloxplot*).
- Pero **ojo**, un outlier **podría ser correcto** aunque sea anómalo estadísticamente (necesidad de un experto).

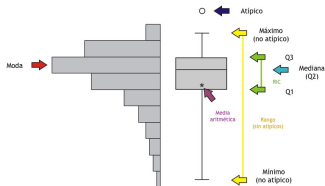


Diagrama de caja con valor outlier o valor atípico.

Ejemplo

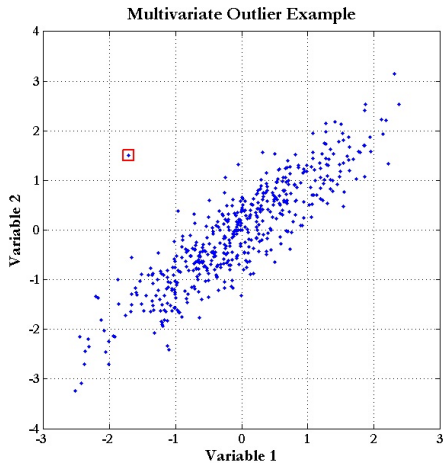


Figura 10: Detección de valores atípicos (outliers) mediante diagrama de dispersión.

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.
 - ▶ $\text{valor} < (Q1 - 1.5\text{IRC})$ o $\text{valor} > (Q3 + 1.5\text{IRC}) \rightarrow$ atípico.
 - ▶ $\text{valor} < (Q1 - 2\text{IRC})$ o $\text{valor} > (Q3 + 2\text{IRC}) \rightarrow$ atípico extremo.
 - ▶ $\text{valor} < (Q1 - 3\text{IRC})$ o $\text{valor} > (Q3 + 3\text{IRC}) \rightarrow$ 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)

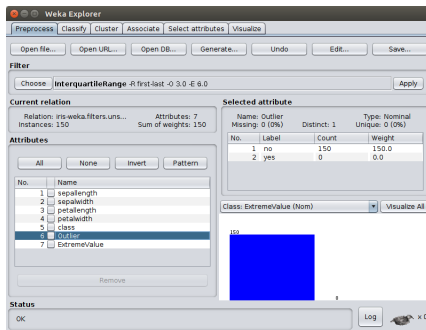


Figura 11: Detección de outliers en el conjunto de datos Iris.

Introducción

Preparación

Visualización

Datos perdidos

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

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.

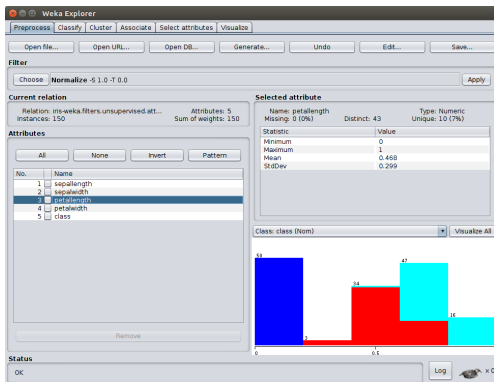


Figura 12: Normalización del conjunto de datos Iris.

Discretización

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

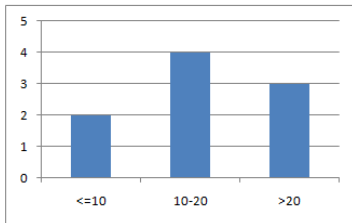
- Por ejemplo, dispongo de una variable edad que toma valores de 5 a 64 años.
- Genero una variable categórica nominal con estas categorías:
 $\{\text{edad} \leq 10, 10 < \text{edad} \leq 30, 30 \leq \text{edad} < 45, \text{edad} \geq 45\}$
- Representar una variable con valores discretos permite **reducir 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

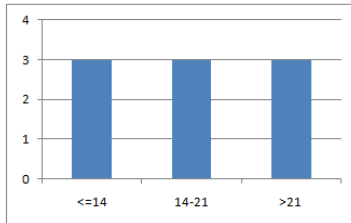
- **Igual amplitud.**

- ▶ 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

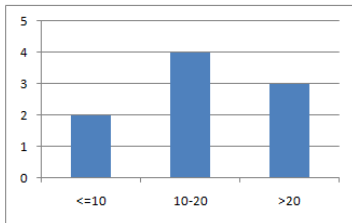


Discretización

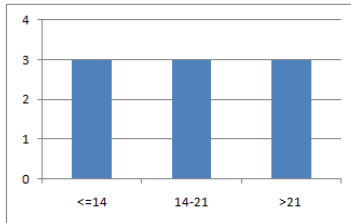
- Igual frecuencia.**

- ▶ 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.

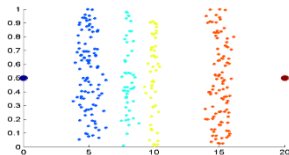
Equal width



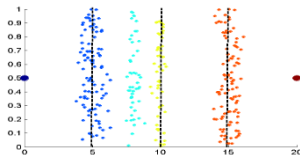
Equal frequency



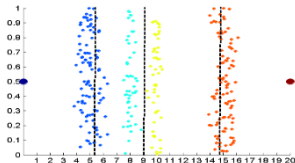
Discretización



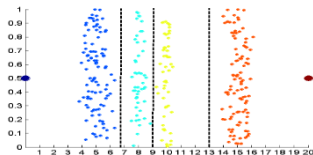
Datos



Igual anchura de intervalo



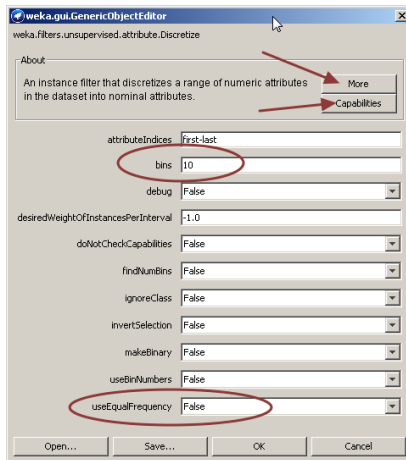
Igual frecuencia



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.

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter: Choose **Discretize-B 10 -M -1.0 -R first-last** Apply

Current relation: Relation: train_iris-weka.filters.unsupervised.attribute.Discretize-B 10... Instances: 111 Attributes: 5

Attributes: All None Invert Pattern

No.	Name
1	sepalength
2	sepalwidth
3	petalength
4	petalwidth
5	Class

Remove

Status: OK Log x 0

Selected attribute: Name: sepalength Missing: 0 (0%) Distinct: 10 Type: Nominal Unique: 0 (0%)

No.	Label	Count
1	'(-inf-4.73]'	7
2	'(4.73-5.06]'	17
3	'(5.06-5.39]'	7
4	'(5.39-5.72]'	23
5	'(5.72-6.05]'	11
6	'(6.05-6.38]'	15
7	'(6.38-6.71]'	16

Class: Class (Nom) Visualize All

Bar chart showing the distribution of sepalength values across different classes. The x-axis represents the sepalength bins, and the y-axis represents the count. The bars are colored blue and red, indicating different classes.

Introducción

Preparación

Visualización

Datos perdidos

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

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 [3].

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

El **análisis de correlaciones** se puede usar también como **método de selección de características**.

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**.

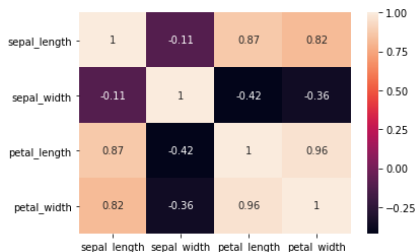


Figura 13: Matriz de correlación (mediante mapa de calor) del conjunto de datos Iris.

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

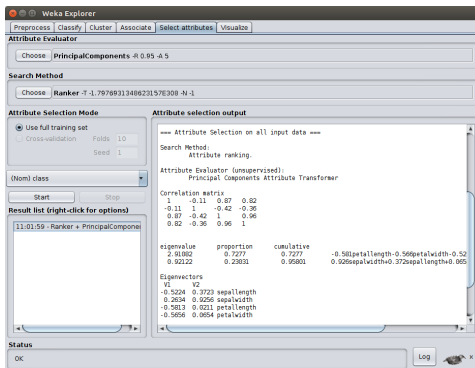


Figura 14: Matriz de correlación para el conjunto de datos Iris.

SC por análisis de correlaciones en Weka (II)

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

Select attributes → CorrelationAttributeEval, Ranker

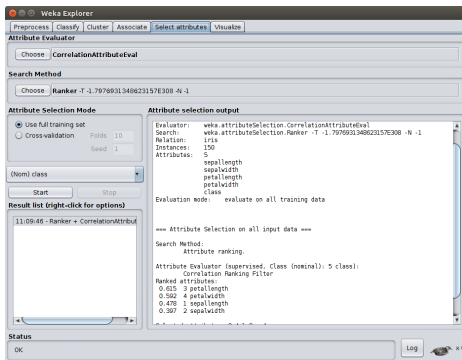
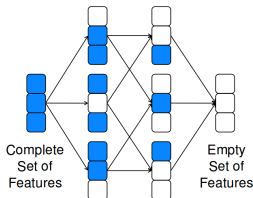


Figura 15: Valoración de cada atributo respecto a la clase en función de su correlación para Iris.

SC mediante búsqueda + evaluación

Subconjuntos de características mediante **búsqueda y 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, con dos componentes:
 - ▶ Una **función de evaluación** que permita comparar dos conjuntos de variables.
 - ▶ Una **estrategia (heurística)** para seleccionar subconjuntos.



SC mediante búsqueda + evaluación

2 métodos según la **evaluación del subconjunto seleccionado**:

- Métodos **Wrapper**: Rendimiento obtenido por un algoritmo de aprendizaje.
 - ▶ Normalmente el que se utilizará posteriormente.
- Métodos **Filter**: Medidas estadísticas
 - ▶ Distancia, correlaciones, información, consistencia, etc.

2 métodos según el resultado obtenido:

- **Individual**: *Ranking* de todos los atributos.
- **Subconjunto**: Subconjunto de atributos.

SC mediante búsqueda + evaluación

Métodos *Wrapper*.

Ventajas:

- subconjuntos pequeños
- buena generalización (interacción con algoritmo de aprendizaje)
- no tienden a sobreentrenar

Desventajas:

- lentos (proceso de aprendizaje)
- introducen sesgo (algoritmo utilizado)
- locales (mal rendimiento con otros algoritmos de aprendizaje)

SC mediante búsqueda + evaluación

Métodos **Filter**.

Ventajas:

- rápidos (cálculo de las medidas)
- globales (independiente del algoritmo de aprendizaje)
- no introducen sesgo

Desventajas:

- subconjuntos con más atributos
- peor resultado con las métricas de acierto

SC mediante búsqueda + evaluación en Weka (I)

Funciones de evaluación → *Attribute Evaluator* en Weka
(nos centraremos en dos).

- ***WrapperSubsetEval***: Evalúa subconjuntos de atributos mediante un algoritmo de aprendizaje.
- ***CfsSubsetEval***: Considera la capacidad de predicción individual de cada atributo junto con el grado de redundancia respecto a los demás.

SC mediante búsqueda + evaluación en Weka (II)

Técnicas de búsqueda → *Search Method* en Weka.

- **Ranker**: *Ranking* de los atributos por sus evaluaciones individuales, luego seleccionar los k mejores.
- **BestFirst**: Método voraz *forward*, *backward*, *bi-directional* y *backtracking*.
- **GreedyStepwise**: Método voraz *forward* y *backward*, permite indicar un subconjunto inicial de atributos.

SC mediante búsqueda + evaluación en Weka (III)

Select attributes → *CfsSubsetEval*, *BestFirst*

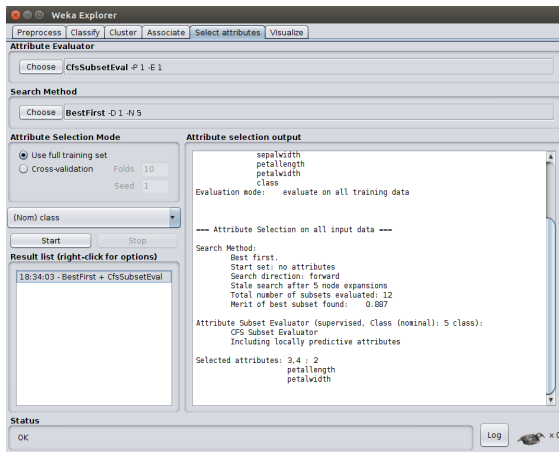


Figura 16: Selección de características *CfsSubsetEval* + *BestFirst* en conjunto de datos Iris.

Introducción

Preparación

Visualización

Datos perdidos

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

Entregables

Bibliografía

Selección de patrones

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

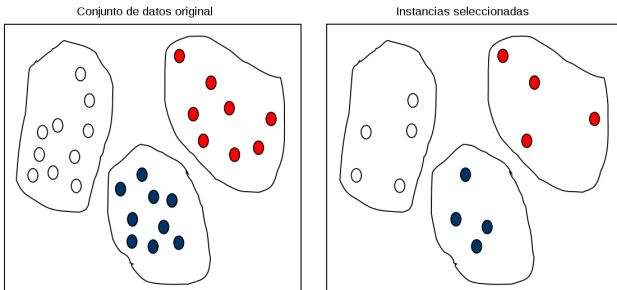


Figura 17: Ejemplo de selección de instancias.

También se emplea para:

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

Selección de patrones 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.
- ***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.

Introducción

Preparación

Visualización

Datos perdidos

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

Entregables

Bibliografía

Conclusiones

El preprocesamiento 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 **preprocesamiento concreto** y diferente del realizado a otros.

Está considerado como un problema desafiante en las tareas de investigación [4].

Introducción

Preparación

Visualización

Datos perdidos

Outliers

Transformación

Sel. características

Sel. instancias

Conclusiones

Entregables







Bibliografía

Entregables

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

1. Describa las operaciones de preprocesamiento que ha realizado sobre la base de datos proporcionada y cómo queda la base de datos final ya preprocesada. 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.

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

-  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.
-  Q. Yang, X. Wu. 10 Challenging problems in data mining research. International Journal of Information Technology and Decision Making 5:4, 597-604., 2006.
-  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?