Práctica 2:

Introducción a WEKA y ANACONDA (python)

UNIVERSIDAD DE VALLADOLID

Técnicas de Aprendizaje Automático
Grado de Ingeniería Informática
Doble Grado con Estadística (INDAT)
Departamento de Informática (ATC, CCIA y LSI)

Datos Crédito

- Crear un archivo con la de datos utilizada en la presentación de la asignatura para ilustrar la inducción de árboles con ID3 (crédito) en los siguientes formatos:
  - CSV
  - ARFF

Árbol de Decisión

- Inducir un árbol de decisión con el algoritmo ID3:
  - Si se desea visualizar gráficamente el árbol, recurrir a J48:
    - collapseTree = False
    - MinNumObj = 1
    - Unprunned = True
- En las opciones de test, elegir las muestras de aprendizaje
- Comprobar si coincide con el resultado de la clase de teoría

Ampliación del árbol de decisión

- Añadir la siguiente muestra al conjunto de aprendizaje:

Nº	Riesgo	Historia	Deuda	Avales	Ingresos
15	bajo	mala	alta	adecuados	0 a 2M

- Crear el nuevo árbol de decisión según la dispositiva anterior
- Comparar los resultados

Ejemplo de función lógica (entrega)

- Tome la siguiente función lógica y obtenga el árbol de decisión (gráfico) correspondiente usando PYTHON.

$$\left(\left(A \wedge B\right) \vee \left(\overline{C} \wedge D\right)\right) xor \left(E \vee \overline{F}\right)$$

- Para ello: `from sklearn.tree import DecisionTreeClassifier, export_graphviz`

- Vuelque la tabla construida a un fichero csv
- Construya el árbol correspondiente con weka
- Incorpore dicho gráfico al final fichero “jupyter-notebook” de esta entrega, así como el obtenido con python.
- Compruebe si son iguales
- Verifique que en ambos casos acierta el 100% de las muestras del experimento

Arbol de Crédito en Python (entrega)

- Añadir al fichero anterior, la implementación de lo hecho en WEKA, pero con Python.
- Aparte de lo usado en la práctica de la semana anterior, se necesita instalar el módulo: decision-tree-id3.

```
import six
import sys
sys.modules['sklearn.externals.six'] = six
```

- No admite datos de tipo cadena, pero sí enteros.
- Por tanto, se hará un transformación de la tabla original:

Transformación de tablas para Id3 en Python

	Riesgo	Historia	Deuda	Avales	Ingresos
0	2	1	1	0	0
1	2	0	1	0	1
2	1	0	0	0	1
3	2	0	0	0	0
4	0	0	0	0	2
5	0	0	0	1	2
6	2	1	0	0	0
7	1	1	0	1	2
8	0	2	0	0	2
9	0	2	1	1	2
10	2	2	1	0	0
11	1	2	1	0	1
12	0	2	1	0	2
13	2	1	1	0	1

	Riesgo	Historia	Deuda	Avales	Ingresos
0	alto	mala	alta	no	0 a 2M
1	alto	desconocida	alta	no	2 a 5M
2	moderado	desconocida	baja	no	2 a 5M
3	alto	desconocida	baja	no	0 a 2M
4	bajo	desconocida	baja	no	más de 5M
5	bajo	desconocida	baja	adecuados	más de 5M
6	alto	mala	baja	no	0 a 2M
7	moderado	mala	baja	adecuados	más de 5M
8	bajo	buena	baja	no	más de 5M
9	bajo	buena	alta	adecuados	más de 5M
10	alto	buena	alta	no	0 a 2M
11	moderado	buena	alta	no	2 a 5M
12	bajo	buena	alta	no	más de 5M
13	alto	mala	alta	no	2 a 5M

Resultado

- No hacer el árbol con el ejemplo adicional correspondiente a la diapositiva 4.
- Obtener los árboles (pdf, jpeg, png, etc.) para id3 y J48
- Este último viene implementado directamente en sklearn.

Ejemplo árbol Id3 /J48