

Fundamentos de Inteligencia Artificial





PhD Wester Edison Zela Moraya

PhD en Computer Science – Inteligencia Artificial por la Universidad Politécnica de Madrid. Master en Ingeniería de Software por la Universidad de Oxford. Master en Análisis Financiero y Económico por la Universidad Complutense de Madrid. Ingeniero de Sistemas de la UNI.

Amplia experiencia profesional en Transformación Digital, Machine Learning, RPAs, Data Science, Metodologías Ágiles, Microservices, gestión económica de proyectos. Docente de Inteligencia Artificial en la Universidad Nacional de Ingeniería.

Director de TI en empresas en Peru y Europa

Consultor de IA y Datos en la SGTD en la PCM

Miembro del AI Connect Program (US Department y Atlantic Council)

Creador de Troomes.com

Temas – Sesión 3

- Árboles de Decisión
- Random Forest
- Ejercicios

Algoritmos de Aprendizaje Supervisado

Algunos algoritmos en el aprendizaje supervisado:

- **Arboles de Decisión**
- **Random Forest**
- Redes Bayesianas
- **Máquinas de Vectores de Soporte (SVM)**
- **Red Neuronal Artificial**
- Aprendizaje profundo (Deep Learning)

Aprendizaje Supervisado - Clasificación

Cuando usamos clasificación, el resultado es una etiqueta, clase o categoría. Es decir, el resultado de la técnica de machine learning que estemos usando será un valor categórico, dentro de un conjunto finito de posibles resultados (número de clases).

Aquí van algunos ejemplos de regresión:

- ✓ Predecir por si una persona tiene o no tiene diabetes
- ✓ Clasificar si un cliente va a darse de baja un servicio de la compañía
- ✓ Predecir si un estudiante va a salir reprobado de un curso en la universidad
- ✓ Predecir que partido político va a ganar las próximas elecciones

Arboles de Decisión

El objetivo de cualquier árbol de decisión es crear un modelo viable que prediga el valor de una variable objetivo en función del conjunto de variables de entrada.

Dos tipos principales:

- El análisis del árbol de clasificación es cuando el resultado predicho es la clase a la que pertenecen los datos.
- El análisis del árbol de regresión es cuando el resultado predicho puede considerarse un número real.

Se utiliza en muchas industrias: instituciones financieras, comercializadores, campo médico y otros

Ventaja: es fácil de leer. Desventaja: puede crear modelos complejos en función de los datos

Diferentes Tipos de Algoritmos

- C4.5: Se basa en el método de obtención de información. Permite que los árboles se utilicen para la clasificación. En Weka está el algoritmo J48.

Calcule el umbral para la división:

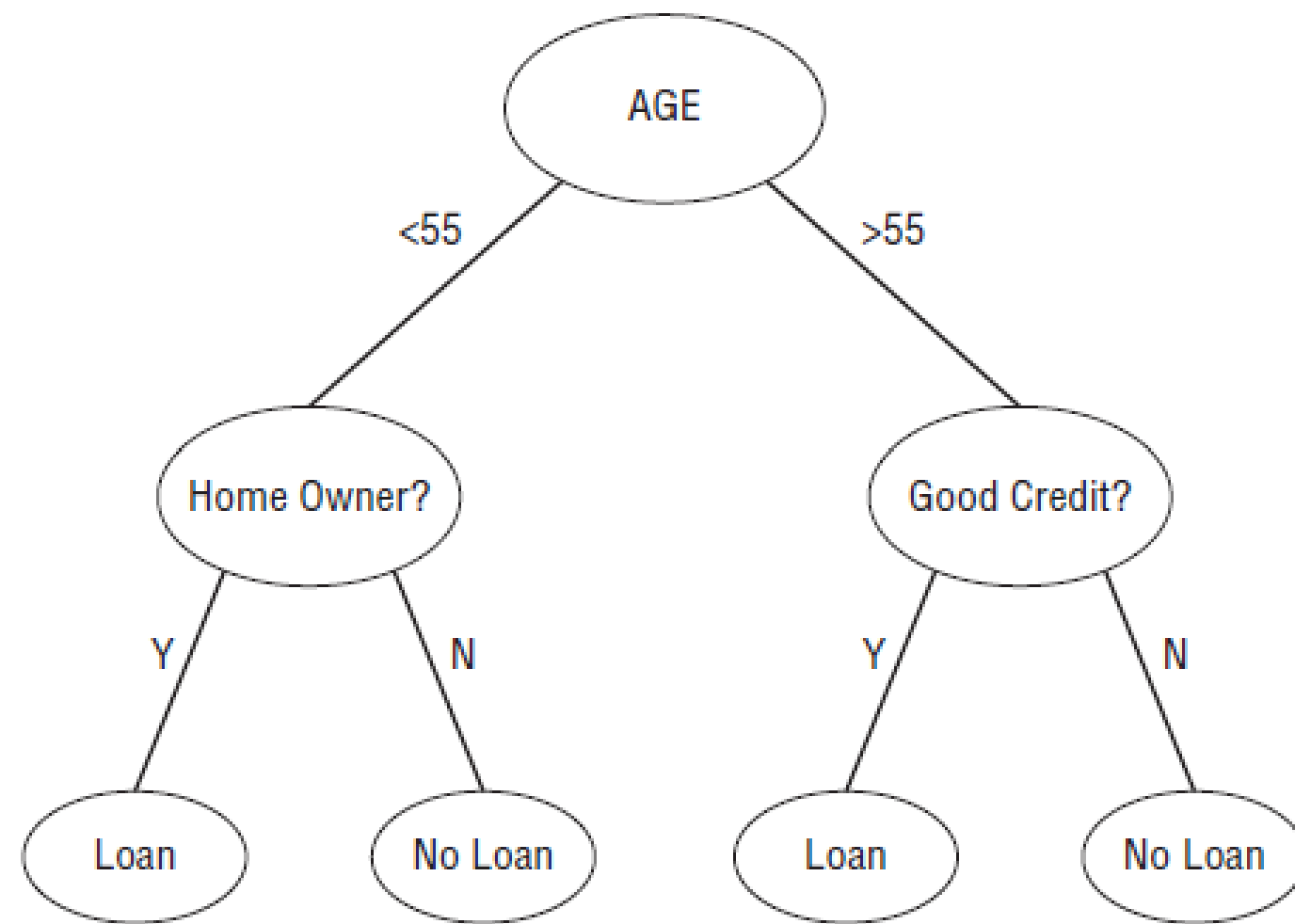
85,80,83,70,68,65,64,72,69,75,75,72,81,71

Dividir y dar un criterio de decisión simple: $a \leq 80$ o $a > 80$

Los árboles se podan

- El ID3 (dicotomizador iterativo 3)
- CHAID (Detección automática de interacción chi-cuadrado)
- MARS (splines de regresión adaptativa multivariante)

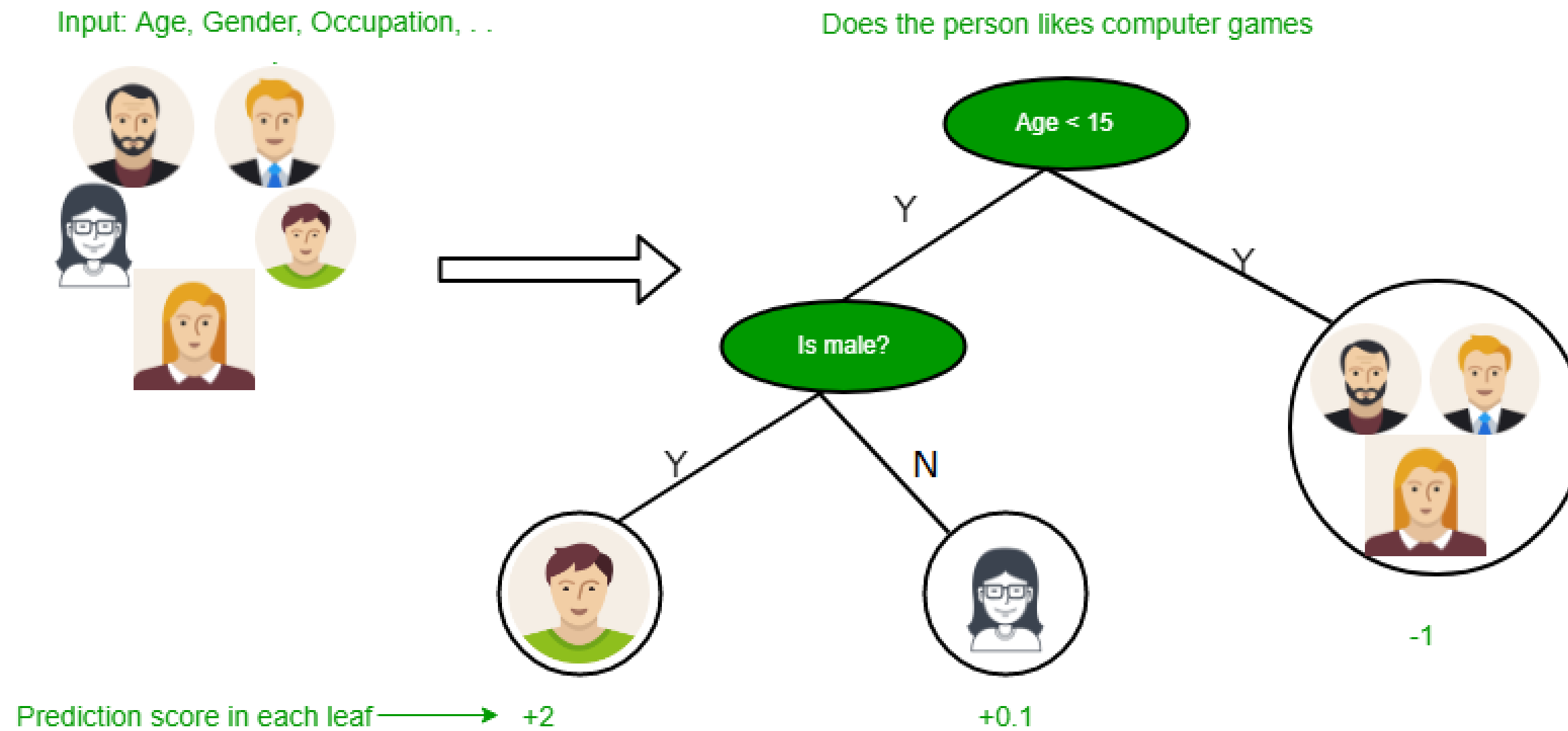
Como los Arboles de Decisión Trabajan



- Los árboles se componen de nodos y bordes
- Cada nodo está asociado con una variable de entrada
- La raíz es el valor total posible de ese nodo
- Una hoja representa un valor basado en los valores dados por la variable de entrada.

¿Cuál es el mejor nodo para iniciar el árbol?

Algorithm for building a Decision Tree



Algoritmo para Construir un Arbol de Decisión

Sigue estos pasos:

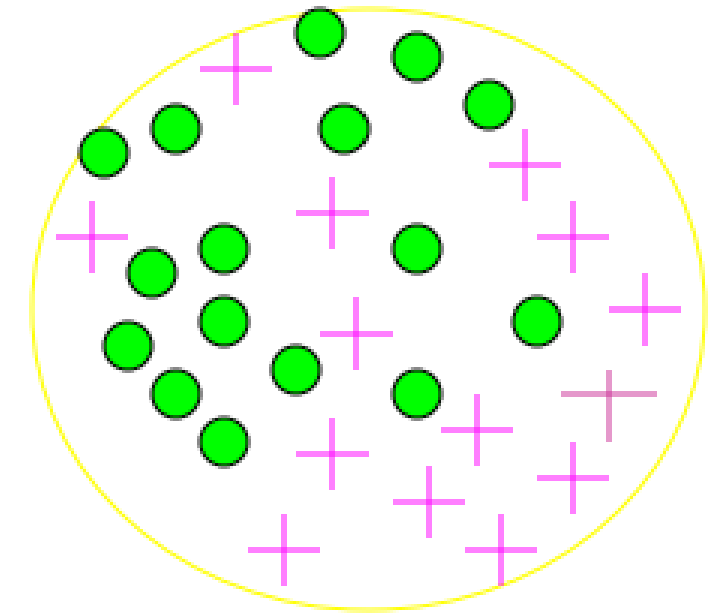
- Verifique el modelo para los casos base.
 - Todas las muestras de la lista pertenecen a la misma clase. simplemente crea un nodo hoja para el árbol de decisiones que dice elegir esa clase.
 - Ninguna de las funciones proporciona ganancia de información. C4.5 crea un nodo de decisión más arriba en el árbol usando el valor esperado de la clase.
 - Se ha encontrado una instancia de una clase nunca antes vista. crea un nodo de decisión más arriba en el árbol usando el valor esperado.
- Repita todos los atributos (attr).
 - Obtenga la ganancia de información normalizada al dividir en attr.
- Hacer de best_attr el atributo con la mayor ganancia de información.
- Cree un nodo de decisión que se divida en el atributo best_attr.
- Trabaje en las sublistas que se obtienen dividiendo en best_attr y agregue esos nodos como nodos secundarios.

Entropía

Cálculo de entropía (medida de incertidumbre).

Tiene un valor [0,1]

$$\text{Entropy} = \sum_i -p_i \log_2 p_i$$



p_i es la probabilidad de la clase i . Calcúlelo como la proporción de la clase i en el conjunto.

16/30 son círculos verdes; 14/30 son cruces rosas. $\log_2 (16/30) = -0,9$; $\log_2 (14/30) = -1,1$ Entropía = $-(16/30) (-0,9) - (14/30) (-1,1) = 0,99$

Entropía = 0: no es bueno para el conjunto de entrenamiento

Entropía = 1: bueno para el conjunto de entrenamiento

Creación de un árbol de decisiones: ejemplo

Ejemplo: historial de compras del usuario:

- ¿Tiene el cliente una cuenta?
- ¿El cliente leyó reseñas de productos anteriores?
- ¿Es el cliente un cliente recurrente?
- ¿El cliente compró el producto?

	HAS CREDIT ACCOUNT?	READ REVIEWS	PREVIOUS CUSTOMER?	DID PURCHASE?
User A	N	Y	Y	Y
User B	Y	Y	Y	Y
User C	N	N	Y	N
User D	Y	N	N	Y
User E	Y	Y	Y	Y

Information Gain

Calcule la Information Gain para casos positivos y negativos

```
package week5;

public class InformationGain {
    private double calcLog2(double value) {
        if(value <= 0.) {
            return 0.;
        }
        return Math.log10(value) / Math.log10(2.);
    }

    public double calcGain(double positive, double negative) {
        double sum = positive + negative;
        double gain = positive * calcLog2(positive/sum)/sum + negative *
            calcLog2(negative/sum)/sum;
        return -gain;
    }

    public static void main(String[] args) {
        InformationGain ig = new InformationGain();
        System.out.println(ig.calcGain(2, 3));
    }
}
```

Calcular Information Gain

Clientes con **credits accounts**: 3 Y y 2 N

$$\begin{aligned}\text{Gain}(3,2) &= (3/5) \cdot \log_2(3/5) + (2/5) \cdot \log_2(2/5) \\ &= 0.97\end{aligned}$$

Los resultados de las variables atributo **read reviews** que se vinculan con las **credit account**

Reads reviews = [Y, Y, N]

Does not read reviews = [N, Y]

$$\text{Gain}(2,1) = (2/3) \cdot \log_2(2/3) + (1/3) \cdot \log_2(1/3) = 0.91$$

$$\text{Gain}(1,1) = (1/2) \cdot \log_2(1/2) + (1/2) \cdot \log_2(1/2) = 1$$

$$\text{Net gain(attribute = has credit account)} = (2/5) \cdot 0.91 + (3/5) \cdot 1 = 0.96$$

Information Gain:

$$\text{InformationGain} = \text{Gain(before the split)} - \text{Gain(after the split)} = 0.97 - 0.96 = 0.01$$

What is the root node?

- The other Attributes:

- Final Table: ***Read Reviews*** should be the root of tree, then ***Is Previous Customer***, and ***Has Credit Account***.

Reads Reviews:

$Gain(3,2) = 0.97$

$Net\ Gain = 0.4$

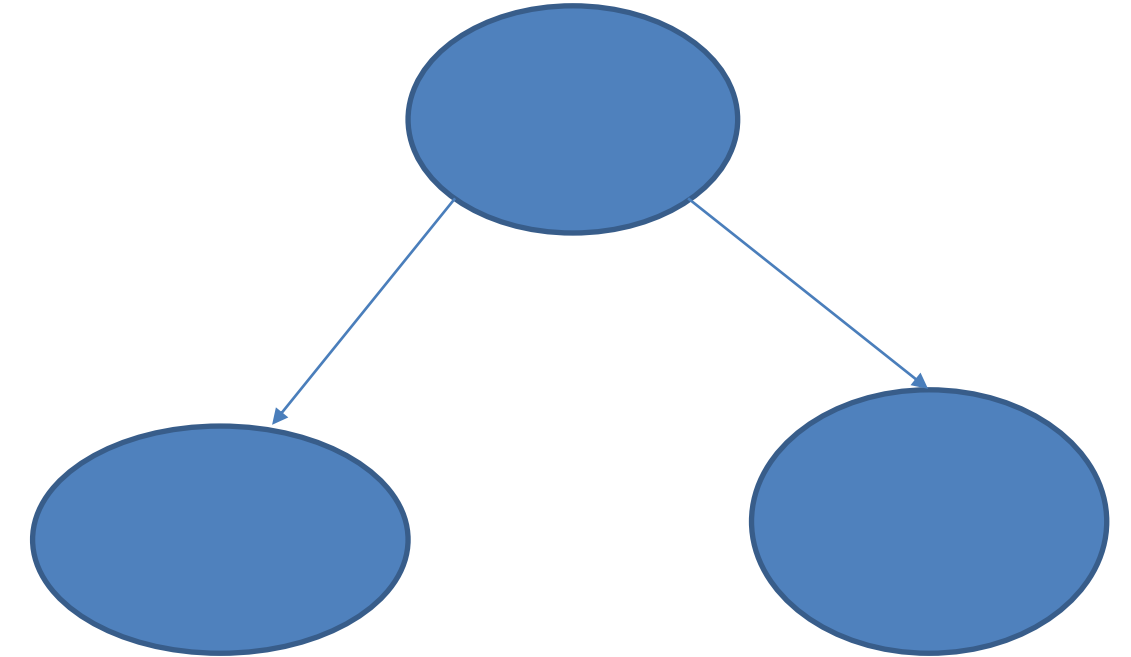
$Information\ Gain = 0.57$

Previous Customer:

$Gain(4,1) = 0.72$

$Net\ Gain = 0.486$

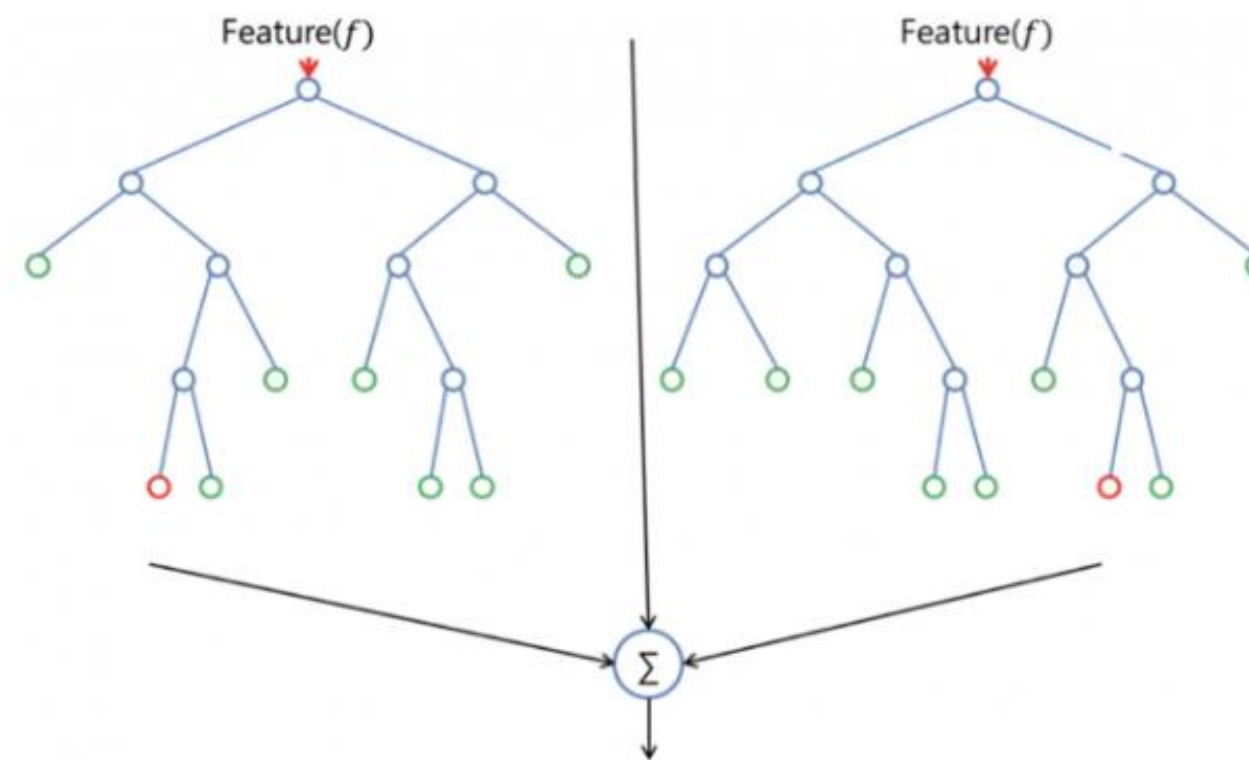
$Information\ Gain = 0.234$



ATTRIBUTE	INFORMATION GAIN
Has Credit Account	0.01
Reads Reviews	0.57
Is Previous Customer	0.234

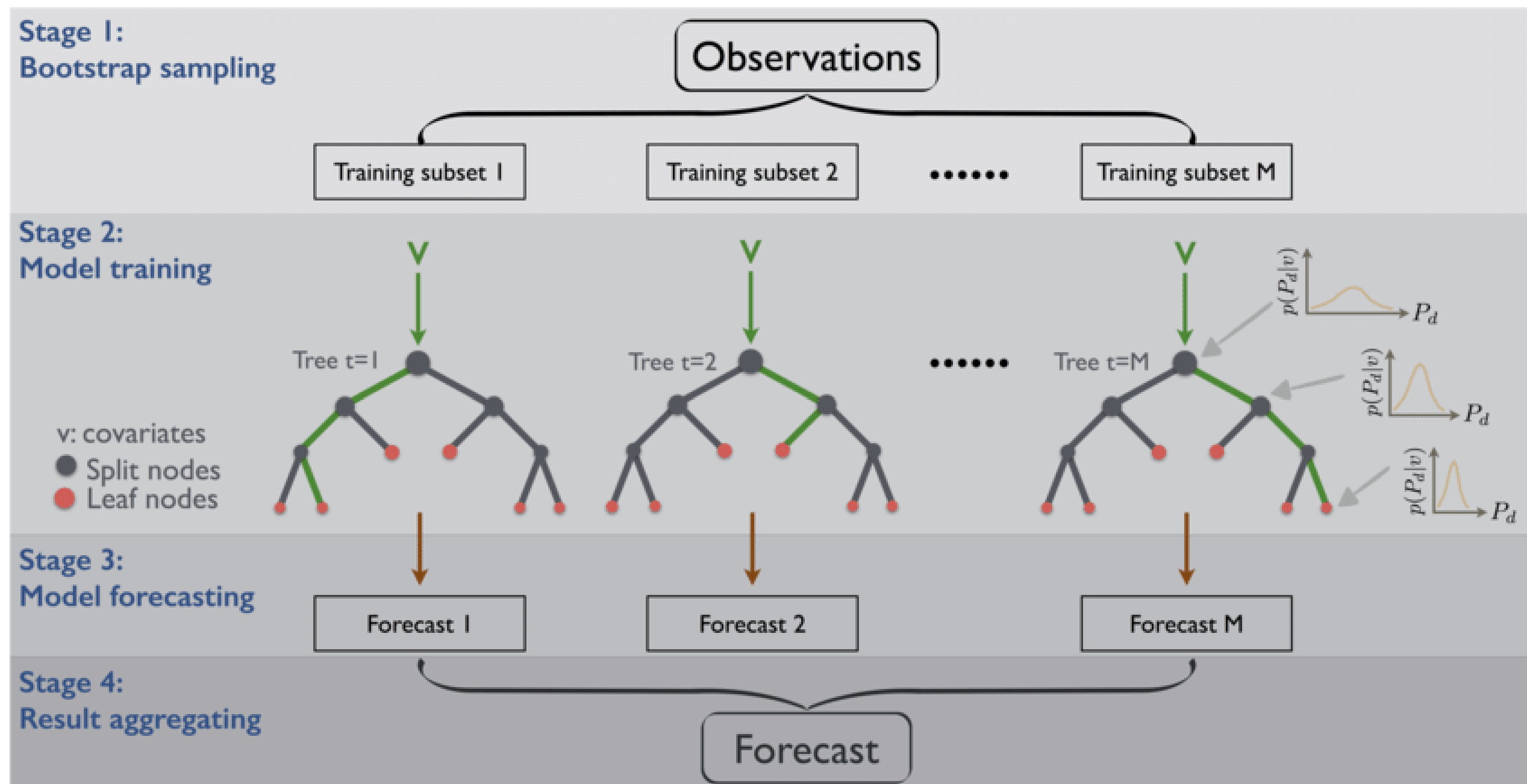
Random Forest

Random Forest es un método de aprendizaje conjunto para clasificación, regresión, que opera mediante la construcción de una multitud de árboles de decisión en el momento del entrenamiento y genera la clase que es el modo de las clases (clasificación) o predicción media (regresión) para arboles individuales.



Random Forest

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners.



Random Forest

El algoritmo de entrenamiento para bosques aleatorios aplica la técnica general de agregación bootstrap, a los aprendices de árboles. Dado un conjunto de entrenamiento $X = x_1, \dots, x_n$ con respuestas $Y = y_1, \dots, y_n$, el ensacado repetido (B veces) selecciona una muestra aleatoria con reemplazo del conjunto de entrenamiento y ajusta los árboles a estas muestras:

Para $b = 1, \dots, B$:

- Muestra, con reemplazo, n ejemplos de entrenamiento de X, Y ; llame a estos X_b, Y_b .
- Entrene un árbol de clasificación o regresión f_b en X_b, Y_b .

Después del entrenamiento, se pueden hacer predicciones para muestras invisibles x' promediando las predicciones de todos los árboles de regresión individuales en x' :

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x')$$

o tomando la mayoría de votos en el caso de árboles de clasificación.

- Evaluation

- Measure the error rate (or performance and switch from one set of features to another one

- Some Performance Evaluation:

- Confussion Matrix
 - Acuraccy
 - Precision/Recall
 - Recieving Operating Characteristics (ROC)

Performance Evaluation

- Feature vectors are used as input for the classifier
- Classification results in a discrete class index
- Confusion matrix:

		hypothesis					
		Ω_1	Ω_2	Ω_3	...	Ω_K	Σ
reference	Ω_1	n_{11}	n_{12}	n_{13}	...	n_{1K}	N_1
	Ω_2	n_{21}	n_{22}	n_{23}	...	n_{2K}	N_2
	Ω_3	n_{31}	n_{32}	n_{33}	...	n_{3K}	N_3
	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
	Ω_K	n_{K1}	n_{K2}	n_{K3}	...	n_{KK}	N_K
Σ							N

Tab.: Confusion matrix with absolute frequencies for a K -class problem

Performance Evaluation

Evaluation of classifiers

- Accuracy / Recognition Rate

$$\text{RR} := \frac{1}{N} \sum_{k=1}^K n_{kk} \cdot 100\%$$

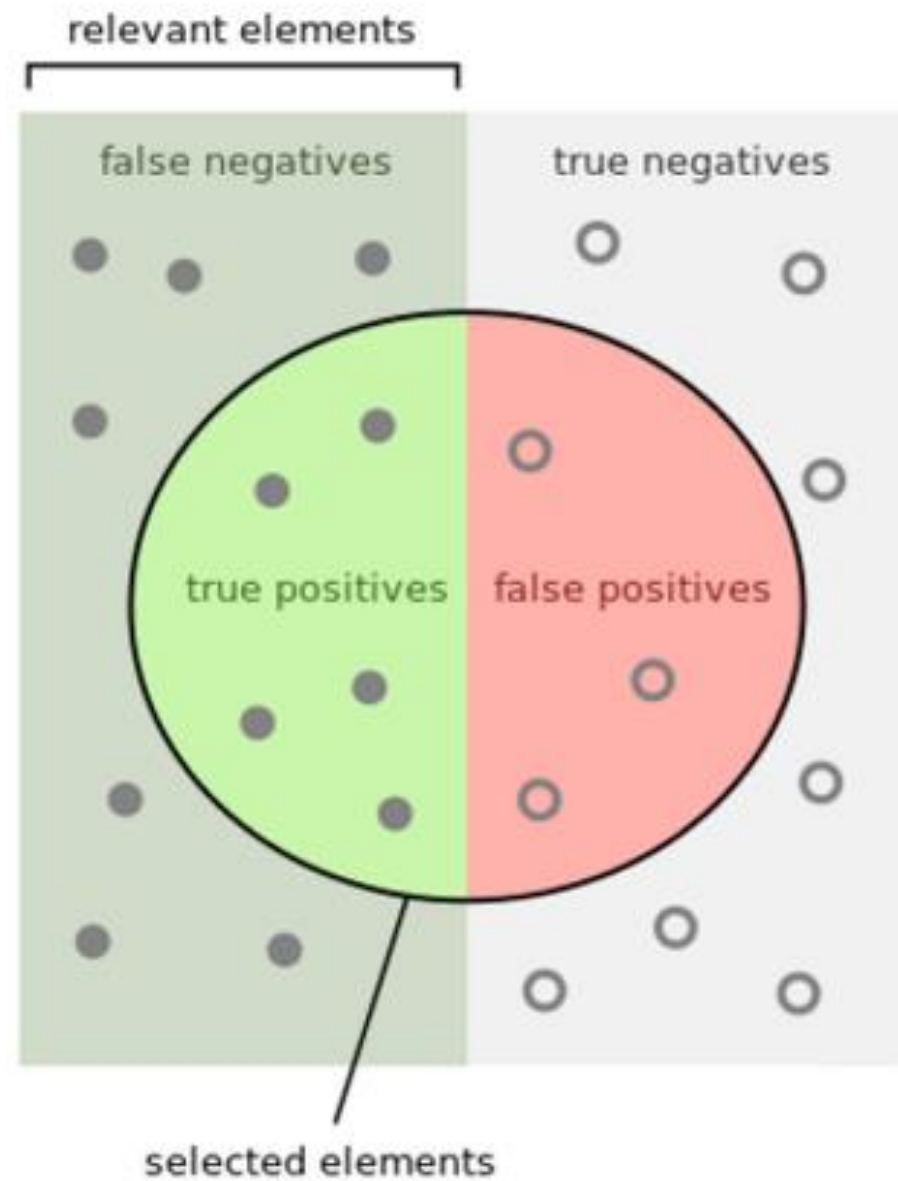
- Recall and Precision


$$\begin{aligned} \text{recall}_k &= \frac{n_{kk}}{\sum_{i=1}^K n_{ki}} = \frac{n_{kk}}{N_k} \\ \text{precision}_k &= \frac{n_{kk}}{\sum_{i=1}^K n_{ik}} \end{aligned}$$


- (Unweighted) Average Recall


$$\text{UAR} := \frac{1}{K} \sum_{k=1}^K \frac{n_{kk}}{N_k} \cdot 100\%$$


Performance Evaluation



Accuracy =  =
$$\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$$

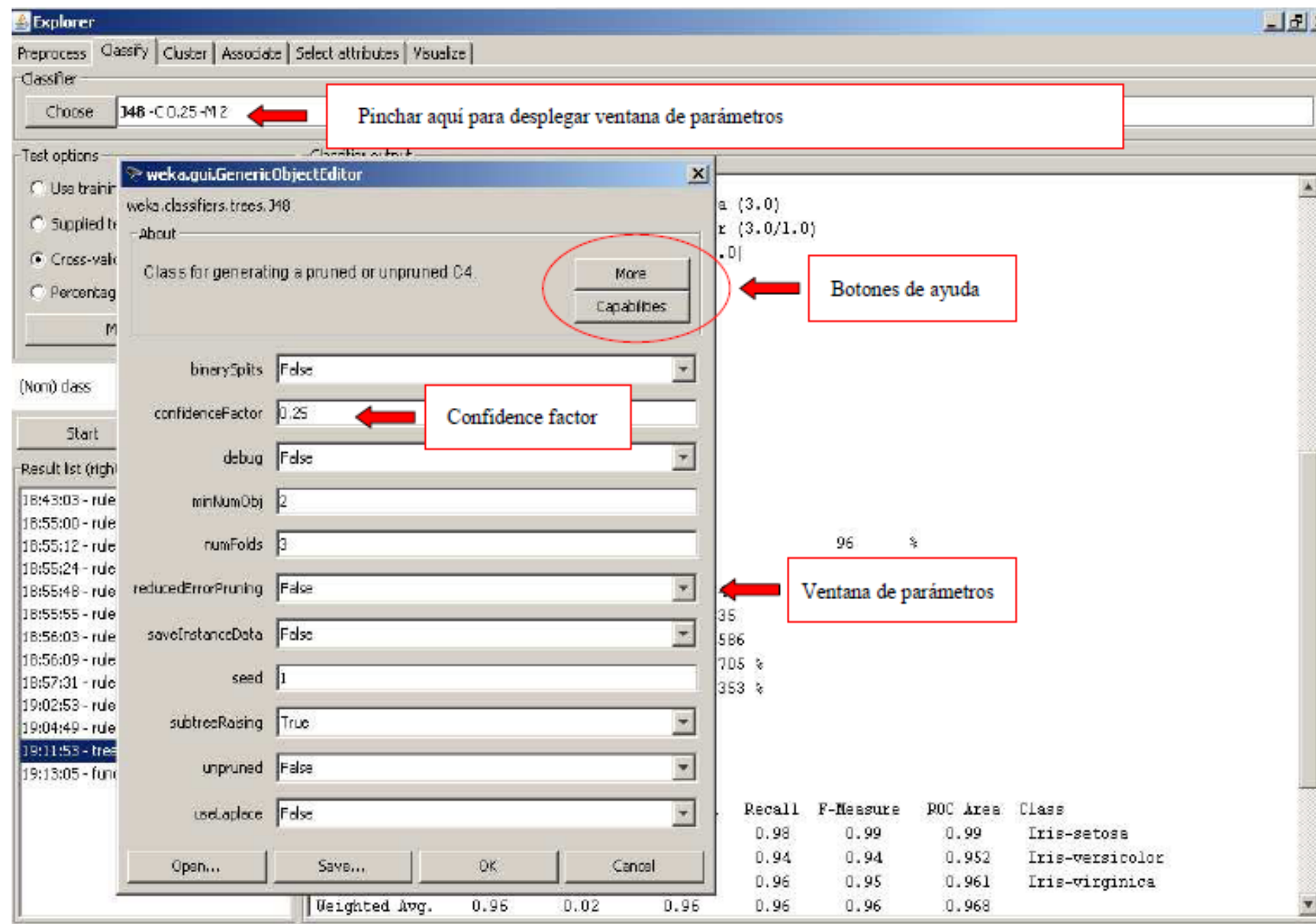
Precision =  =
$$\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$$

Recall =  =
$$\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$$

False Positive Rate = (Fall-out) =  =
$$\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$$

Parámetros del Clasificador

- Seleccionar los parámetros según Clasificador y precisión



Confidence Factor: el factor de confianza utilizado para la poda (los valores más pequeños provocan una mayor poda).

Cuanto menor sea el factor de confianza, más poda hará el algoritmo

La poda es una forma de reducir el tamaño del árbol de decisión.

Si aumenta la poda, la precisión en el conjunto de entrenamiento será menor

Seleccionar el Clasificador J48

- Seleccionar clasificador J48 (Arbol de Decisión)

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize Auto-WEKA DI4j Inference

Classifier

Choose **J48 -C 0.25 -M 2**

Test options

☐ Use training set
☐ Supplied test set Set...
☒ Cross-validation Folds **10**
☐ Percentage split % 66

More options...

(Nom) Churn

Start Stop

Result list (right-click for options)

18:46:11 - trees.J48

Classifier output

Time taken to build model: 0.17 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	3145	94.3594 %
Incorrectly Classified Instances	188	5.6406 %
Kappa statistic	0.7524	
Mean absolute error	0.083	
Root mean squared error	0.2307	
Relative absolute error	33.4618 %	
Root relative squared error	65.5264 %	
Total Number of Instances	3333	

=== Detailed Accuracy By Class ===

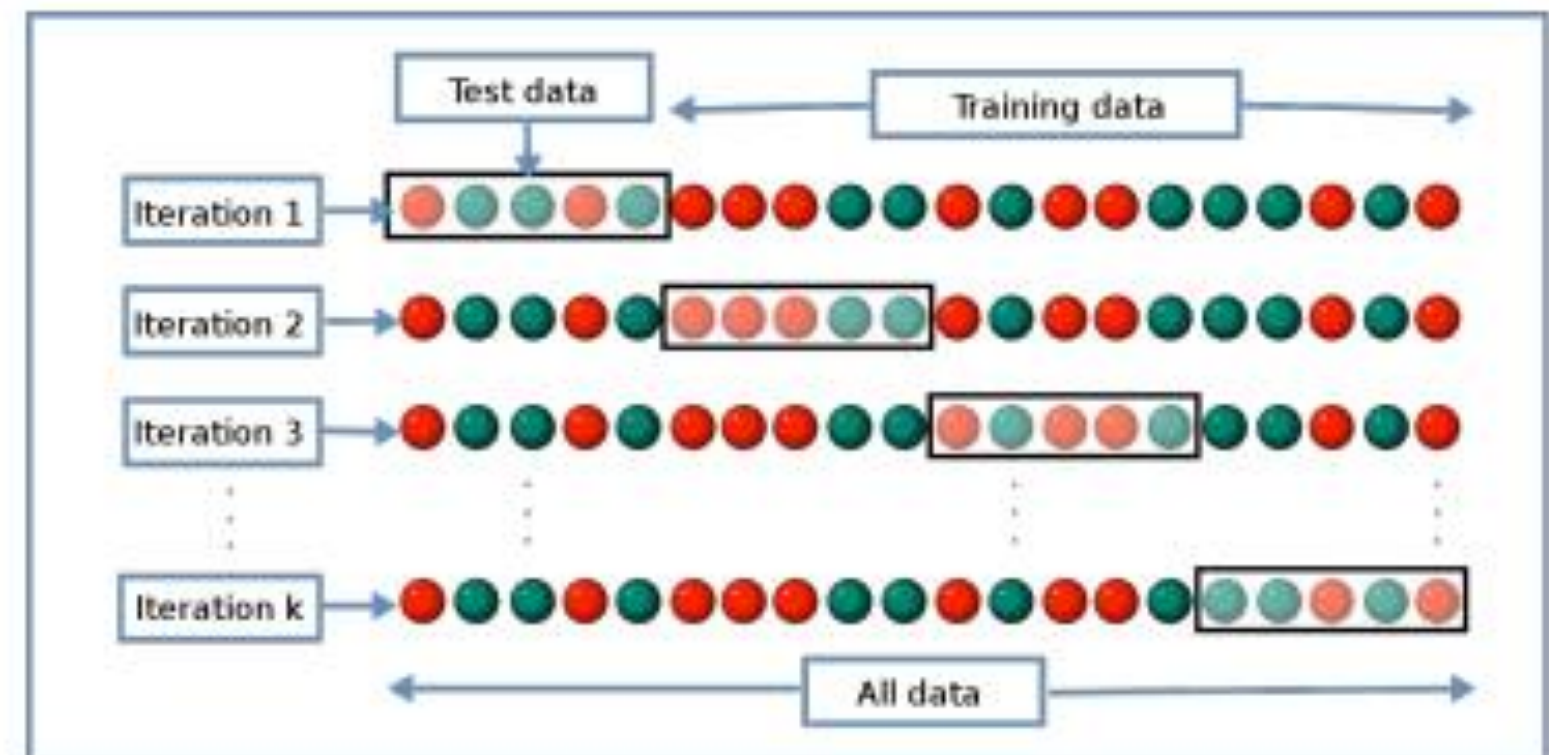
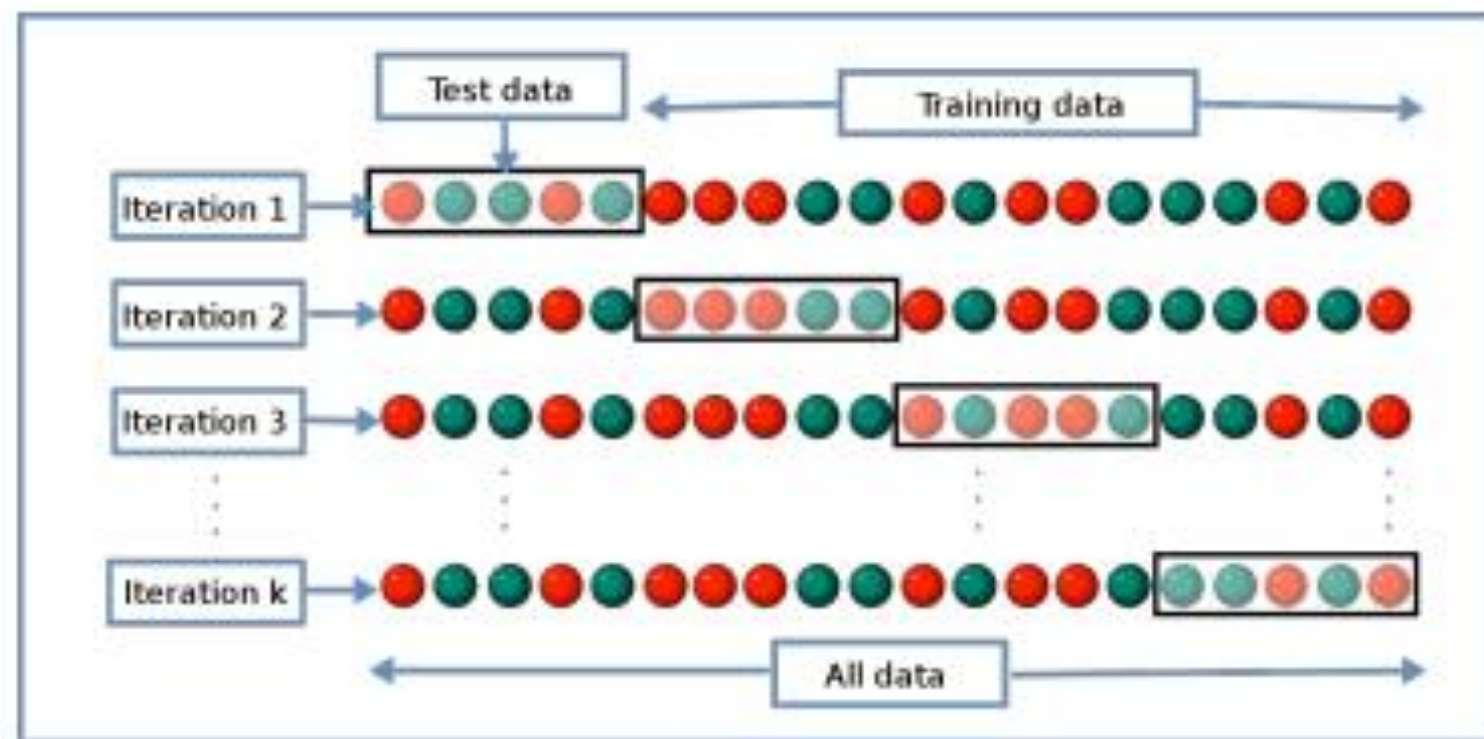
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.984	0.292	0.952	0.984	0.968	0.758	0.854	0.942	No
	0.708	0.016	0.879	0.708	0.784	0.758	0.854	0.730	Yes
Weighted Avg.	0.944	0.252	0.942	0.944	0.941	0.758	0.854	0.912	

=== Confusion Matrix ===

a	b	<-- classified as
2803	47	a = No
141	342	b = Yes

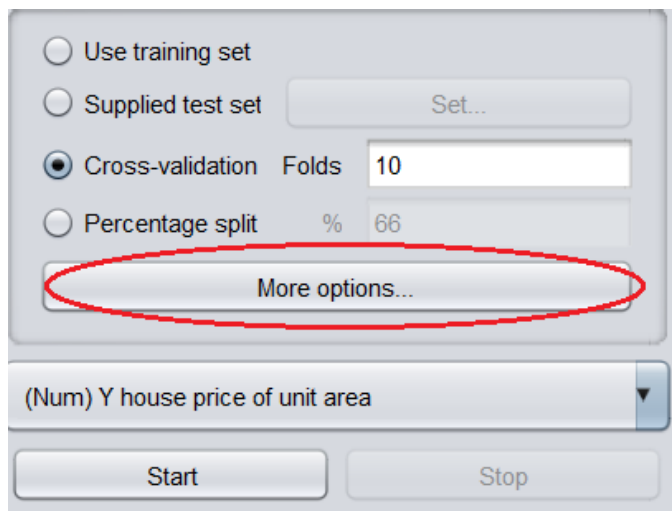
Opciones de Prueba: Cross-Validation k Folds

1. Mezcle el conjunto de datos de forma aleatoria.
2. Divida el conjunto de datos en k grupos
3. Para cada grupo único:
 - a. Tome el grupo como un conjunto de datos de prueba o de reserva
 - b. Tome los grupos restantes como un conjunto de datos de entrenamiento
 - c. Coloque un modelo en el conjunto de entrenamiento y evalúelo en el conjunto de prueba
 - d. Conserve la puntuación de la evaluación y descarte el modelo
4. Resuma la habilidad del modelo usando la muestra de puntajes de evaluación del modelo.

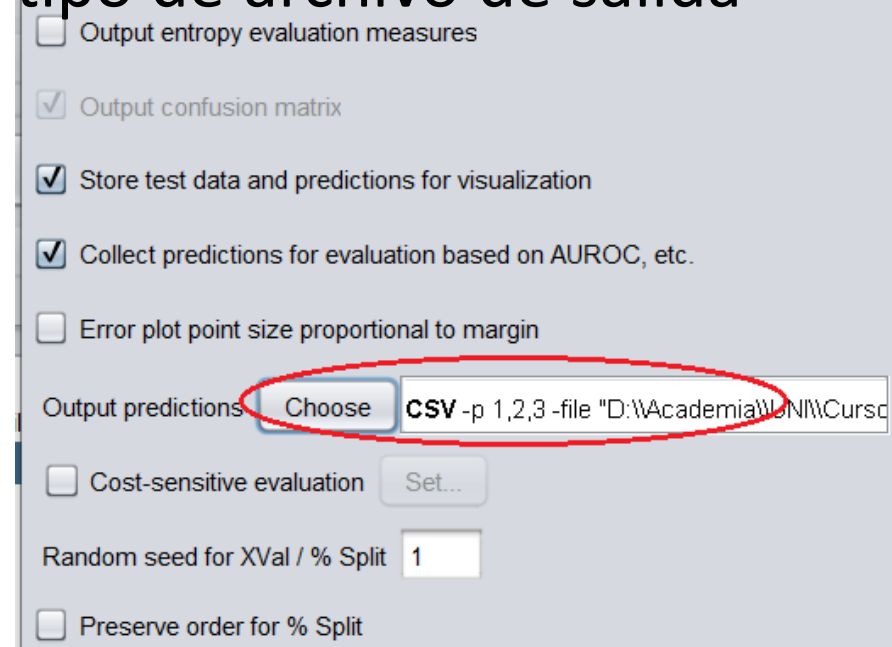


Generación de Resultados de los Modelos Creados

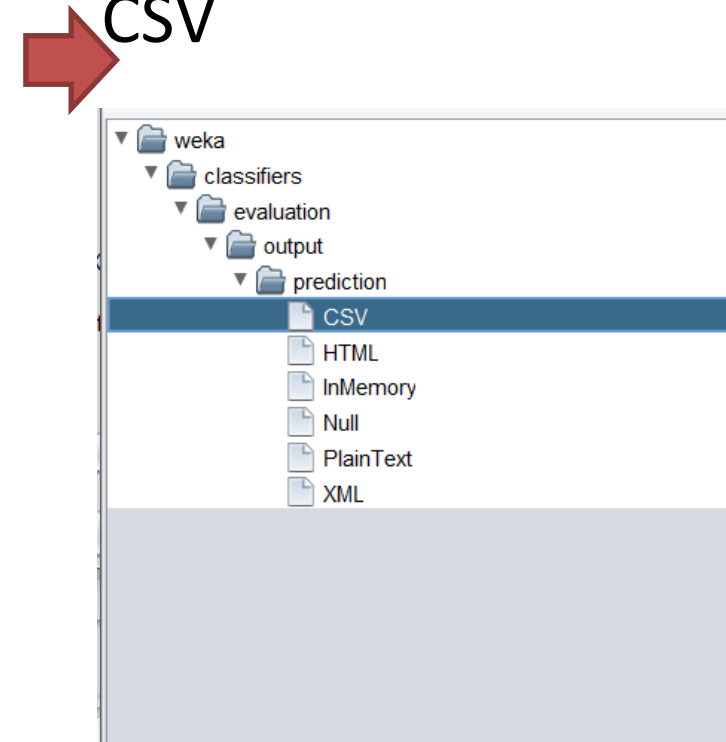
1. Click en **More Options**



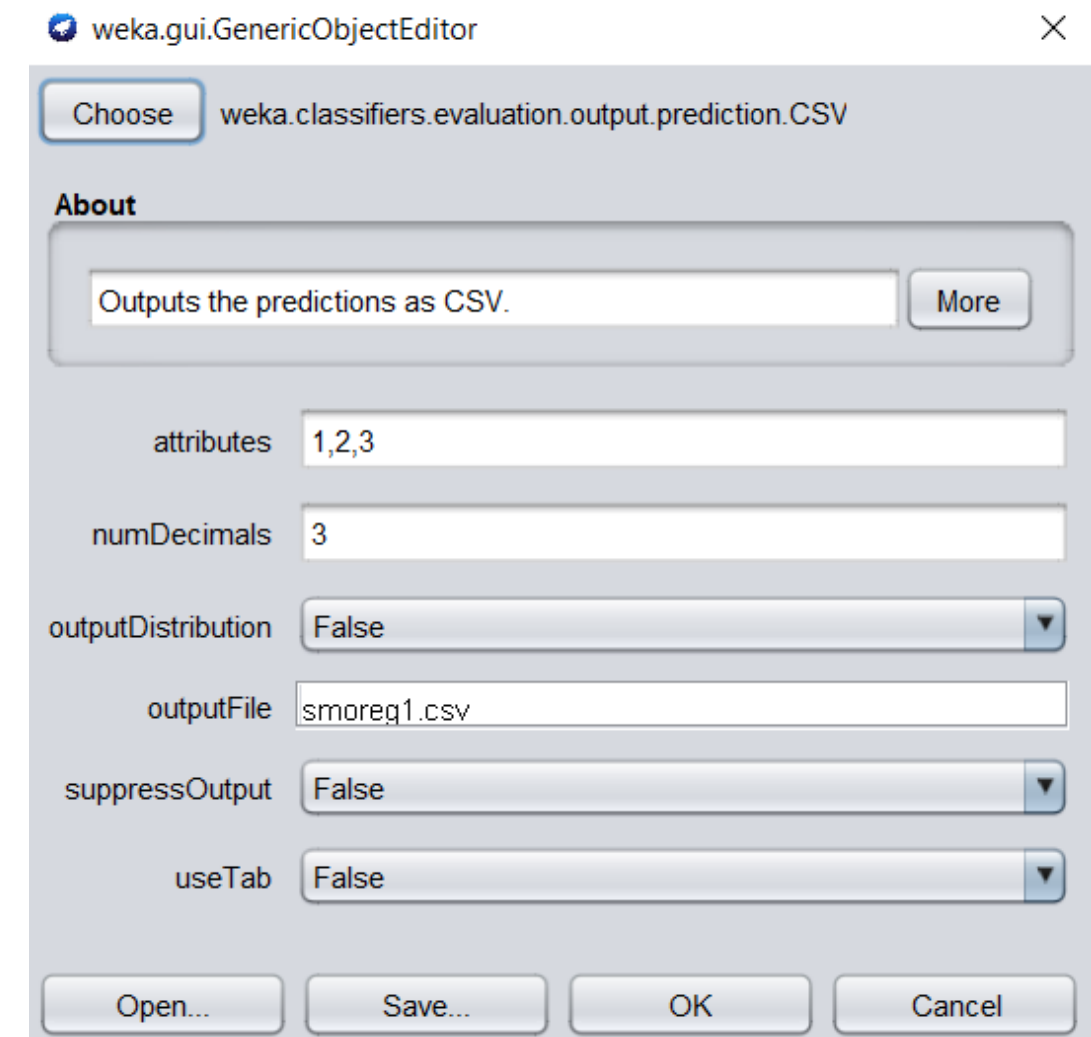
2. Seleccionar Choose, y doble click para seleccionar el tipo de archivo de salida



3. Selección el Tipo de Archivo. Por ejemplo: CSV

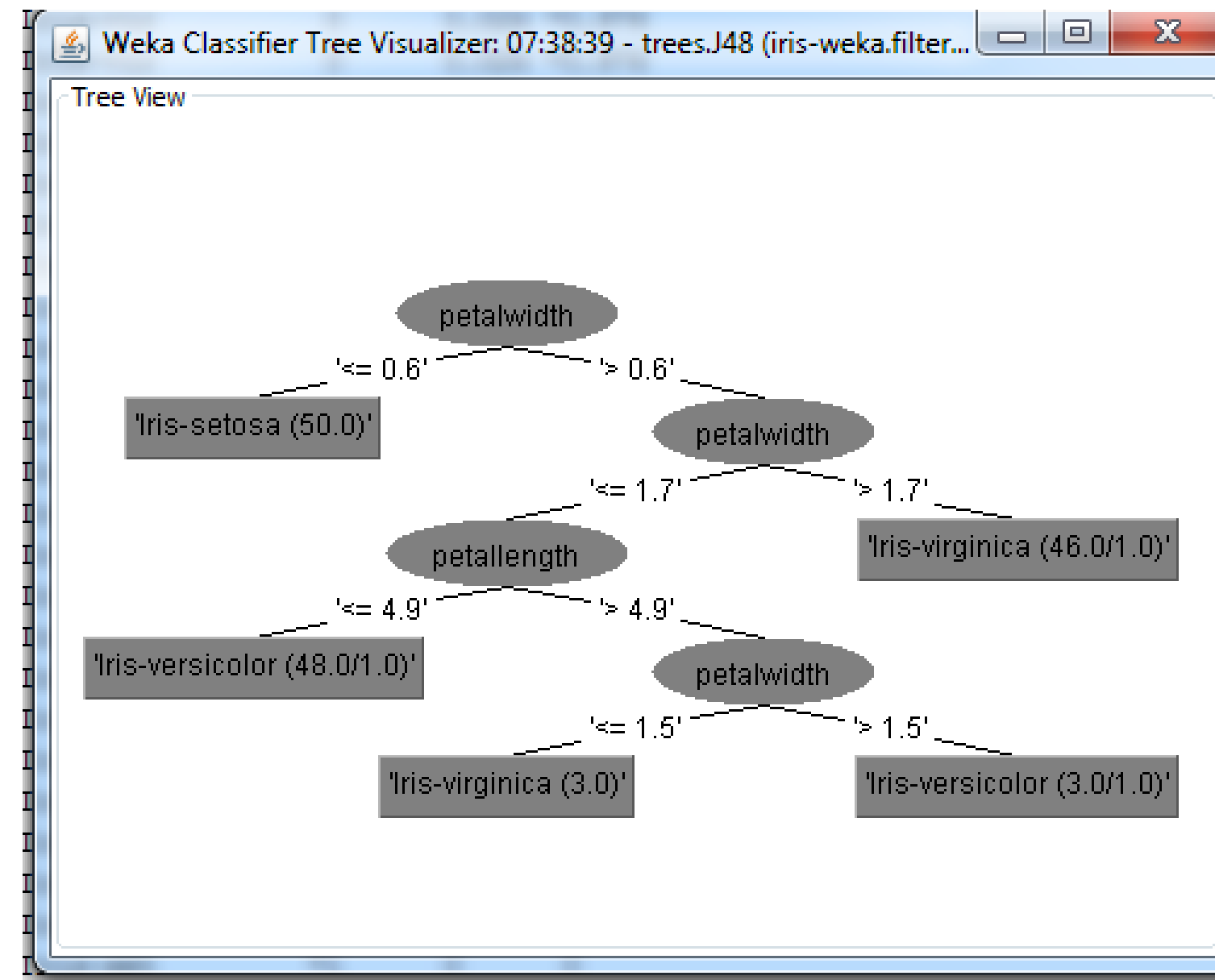


4. Ingresar el nombre del Archivo de Salida



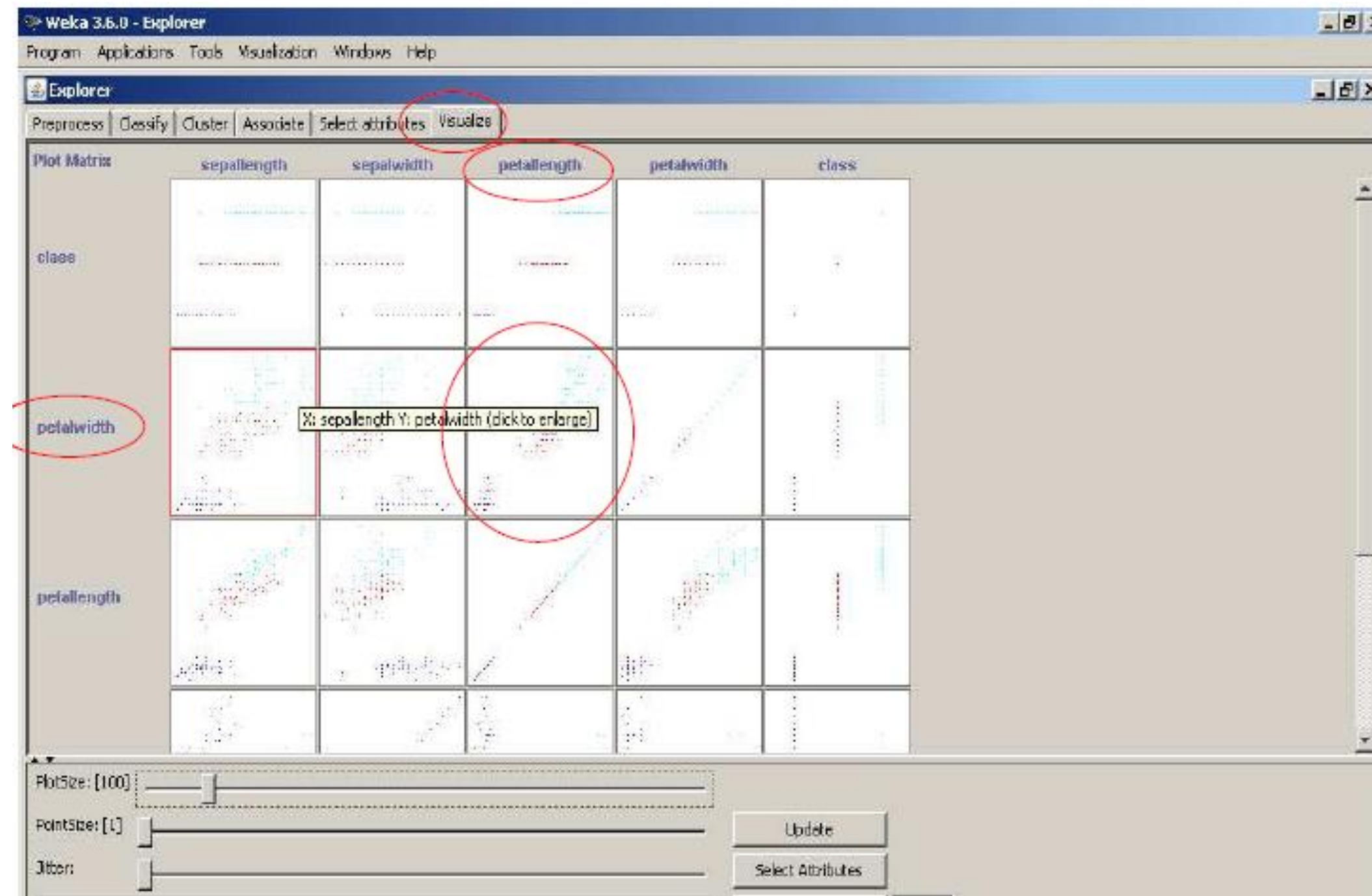
Visualizar el Arbol

- Arbol que ha sido generado con el J48



Visualizar Atributos

- Los atributos mas relevantes: pueden clasificar las clases



Ejercicio: Comparar J48 y Random Forest

Archivo: debiates.arff

- **Número de embarazos**
- **Concentración de glucosa en plasma a 2 horas en una prueba de tolerancia a la glucosa oral**
- **Presión arterial diastólica (mm Hg):** Cuando su corazón está en reposo, entre latidos, su presión arterial baja
- **Espesor del pliegue cutáneo del tríceps (mm)**
- **Insulina sérica de 2 horas (mu U / ml):** Es una prueba que mide cuánta insulina tiene en la sangre.
- **Índice de masa corporal (peso en kg / (altura en m) ^ 2)**
- **Función pedigrí de la diabetes:** Una función que califica la probabilidad de diabetes según los antecedentes familiares.
- **Edad (años)**
- **Variable de clase (0 o 1):** El paciente muestra signos de diabetes (1), (0) en caso contrario.

2.8. Casos de Uso de Clasificación

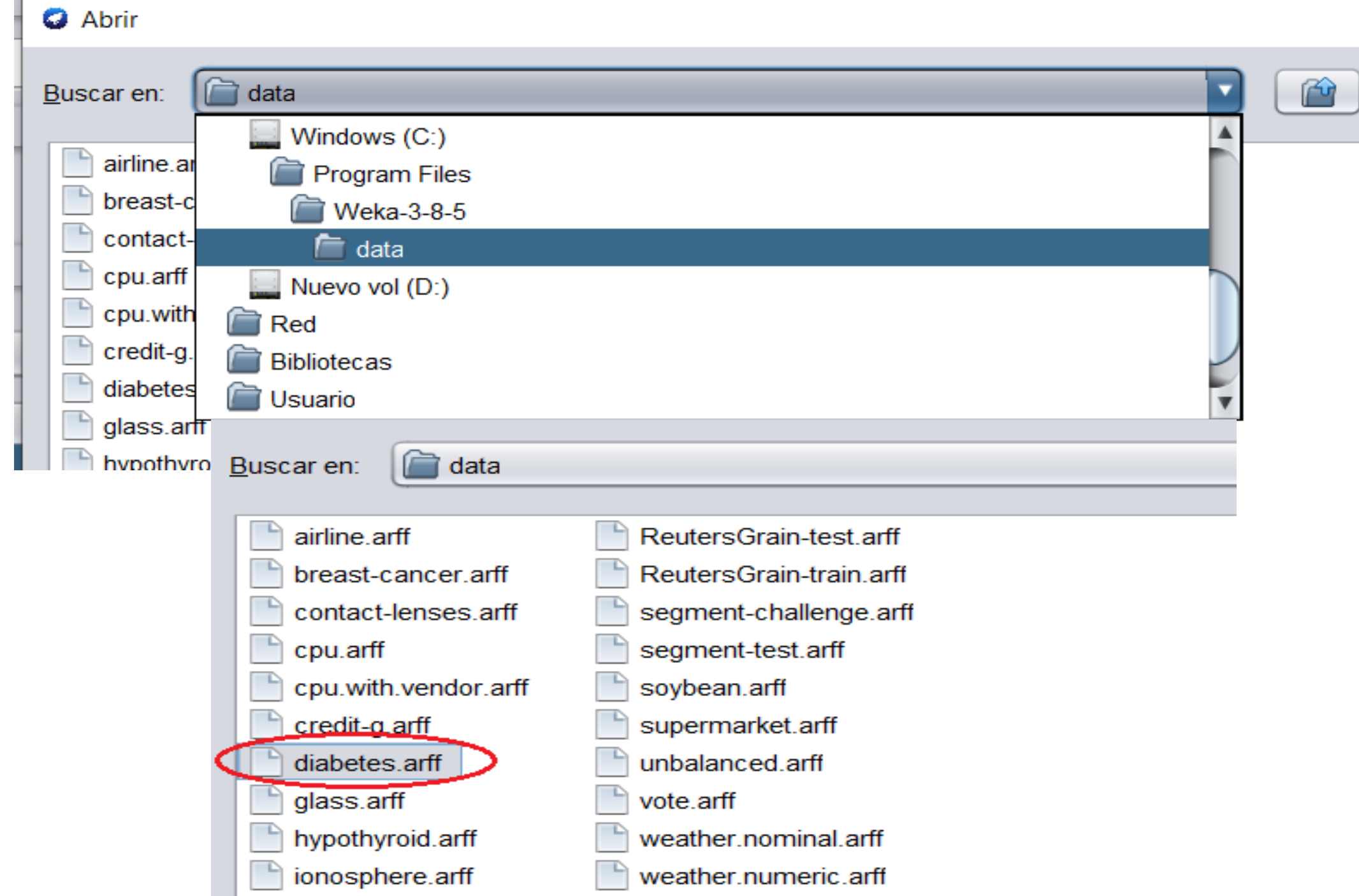
Archivo: debiates.arff

La variable diagnóstica de valor binario investigada representa si el paciente muestra signos de diabetes de acuerdo con los criterios de la Organización Mundial de la Salud (es decir, si la glucosa en plasma 2 horas después de la carga fue de al menos 200 mg / dl en cualquier examen de la encuesta o si se encontró durante la atención médica de rutina). La muestra es de la población vive cerca de Phoenix, Arizona, EE. UU.



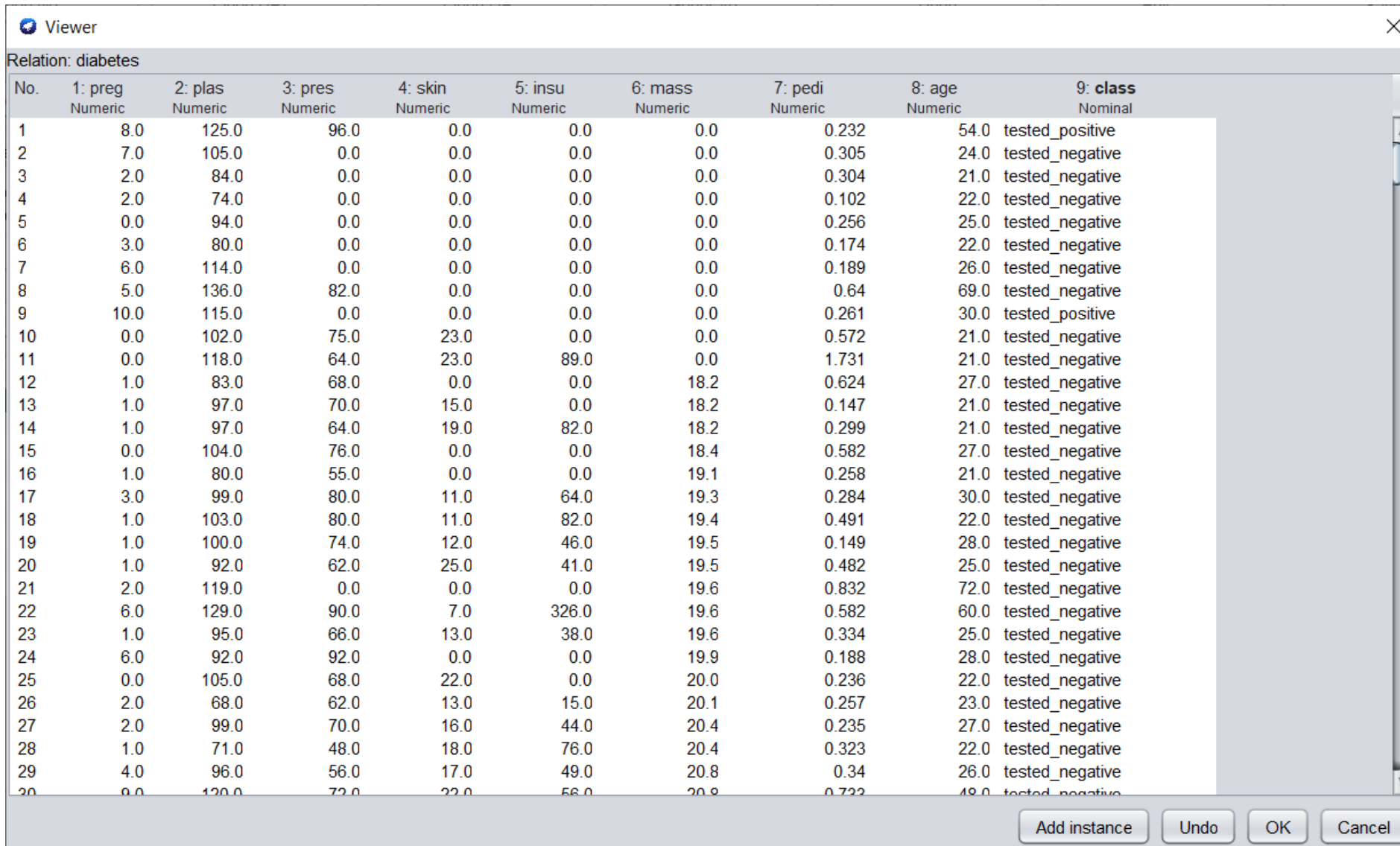
Para Abrir el archivo, en Weka seleccionar Tab Preprocess -> Open file...

El archivo se encuentra en la carpeta de Weka sub carpeta Data



2.9. Casos de Uso de Clasificación

En Tab **Preprocess** -> **Edit**



No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: class Nominal
1	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
2	7.0	105.0	0.0	0.0	0.0	0.0	0.305	24.0	tested_negative
3	2.0	84.0	0.0	0.0	0.0	0.0	0.304	21.0	tested_negative
4	2.0	74.0	0.0	0.0	0.0	0.0	0.102	22.0	tested_negative
5	0.0	94.0	0.0	0.0	0.0	0.0	0.256	25.0	tested_negative
6	3.0	80.0	0.0	0.0	0.0	0.0	0.174	22.0	tested_negative
7	6.0	114.0	0.0	0.0	0.0	0.0	0.189	26.0	tested_negative
8	5.0	136.0	82.0	0.0	0.0	0.0	0.64	69.0	tested_negative
9	10.0	115.0	0.0	0.0	0.0	0.0	0.261	30.0	tested_positive
10	0.0	102.0	75.0	23.0	0.0	0.0	0.572	21.0	tested_negative
11	0.0	118.0	64.0	23.0	89.0	0.0	1.731	21.0	tested_negative
12	1.0	83.0	68.0	0.0	0.0	18.2	0.624	27.0	tested_negative
13	1.0	97.0	70.0	15.0	0.0	18.2	0.147	21.0	tested_negative
14	1.0	97.0	64.0	19.0	82.0	18.2	0.299	21.0	tested_negative
15	0.0	104.0	76.0	0.0	0.0	18.4	0.582	27.0	tested_negative
16	1.0	80.0	55.0	0.0	0.0	19.1	0.258	21.0	tested_negative
17	3.0	99.0	80.0	11.0	64.0	19.3	0.284	30.0	tested_negative
18	1.0	103.0	80.0	11.0	82.0	19.4	0.491	22.0	tested_negative
19	1.0	100.0	74.0	12.0	46.0	19.5	0.149	28.0	tested_negative
20	1.0	92.0	62.0	25.0	41.0	19.5	0.482	25.0	tested_negative
21	2.0	119.0	0.0	0.0	0.0	19.6	0.832	72.0	tested_negative
22	6.0	129.0	90.0	7.0	326.0	19.6	0.582	60.0	tested_negative
23	1.0	95.0	66.0	13.0	38.0	19.6	0.334	25.0	tested_negative
24	6.0	92.0	92.0	0.0	0.0	19.9	0.188	28.0	tested_negative
25	0.0	105.0	68.0	22.0	0.0	20.0	0.236	22.0	tested_negative
26	2.0	68.0	62.0	13.0	15.0	20.1	0.257	23.0	tested_negative
27	2.0	99.0	70.0	16.0	44.0	20.4	0.235	27.0	tested_negative
28	1.0	71.0	48.0	18.0	76.0	20.4	0.323	22.0	tested_negative
29	4.0	96.0	56.0	17.0	49.0	20.8	0.34	26.0	tested_negative
30	0.0	120.0	72.0	22.0	56.0	20.8	0.722	48.0	tested_negative

Los valores de los dataset diabetes.arff

Atributos:

- Preg
- Plas
- Pres
- Skin
- Insu
- Mass
- Pedi
- Age

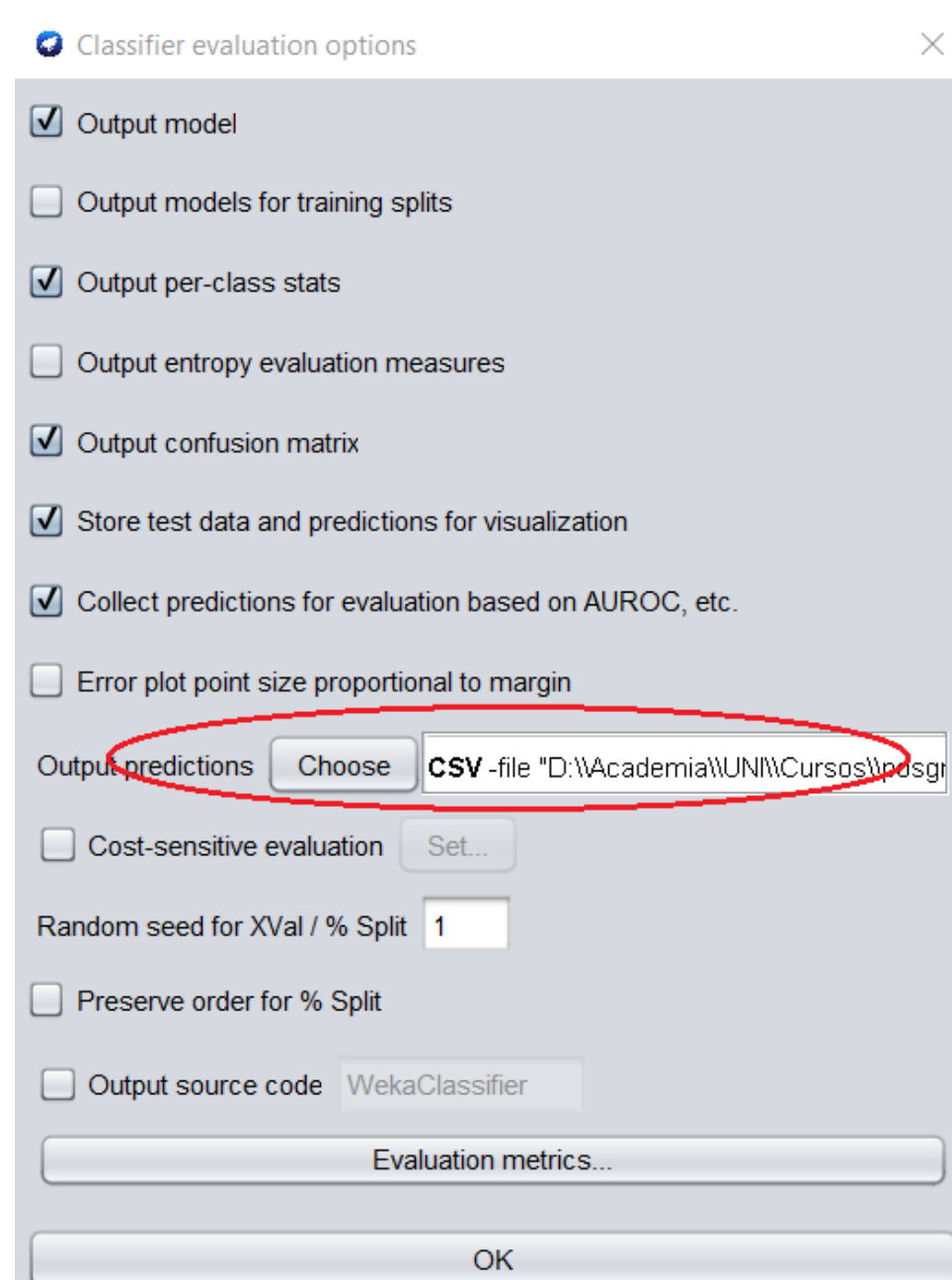
Variable Objetivo :

- Class: tested_positive, tested_negative

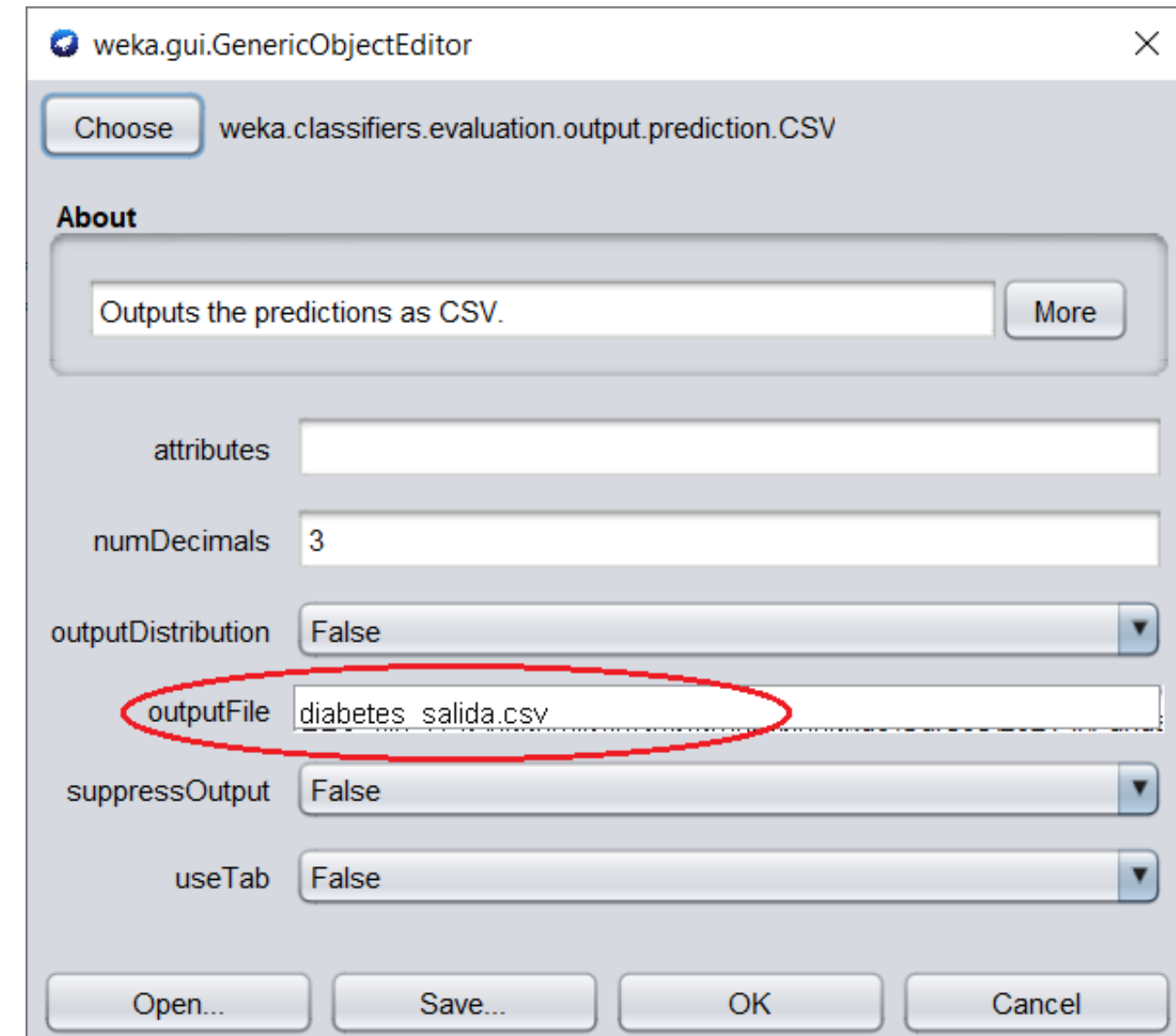
Casos de Uso de Clasificación – Arbol de Decisión

Grabar las salidas de las predicciones realizadas con el modelo entrenado a un archivo de Salida. En **Test Options** -> click **More Options**

Click en **Choose**



Escribir la Ruta donde se grabarán las predicciones realizadas por el modelo



Ejercicio: Comparar J48 y Random Forest

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize Auto-WEKA DI4j Inference

Classifier

Choose **RandomForest** -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

☐ Use training set
☐ Supplied test set Set...
☒ Cross-validation Folds **10**
☐ Percentage split % 66
More options...

(Nom) class

Start Stop

Result list (right-click for options)

- 18:46:11 - trees.J48
- 19:00:21 - trees.J48
- 19:11:58 - trees.J48
- 19:13:01 - trees.RandomForest**

Classifier output

```
weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities
```

Time taken to build model: 0.3 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	582	75.7813 %
Incorrectly Classified Instances	186	24.2188 %
Kappa statistic	0.4566	
Mean absolute error	0.3106	
Root mean squared error	0.4031	
Relative absolute error	68.3405 %	
Root relative squared error	84.5604 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

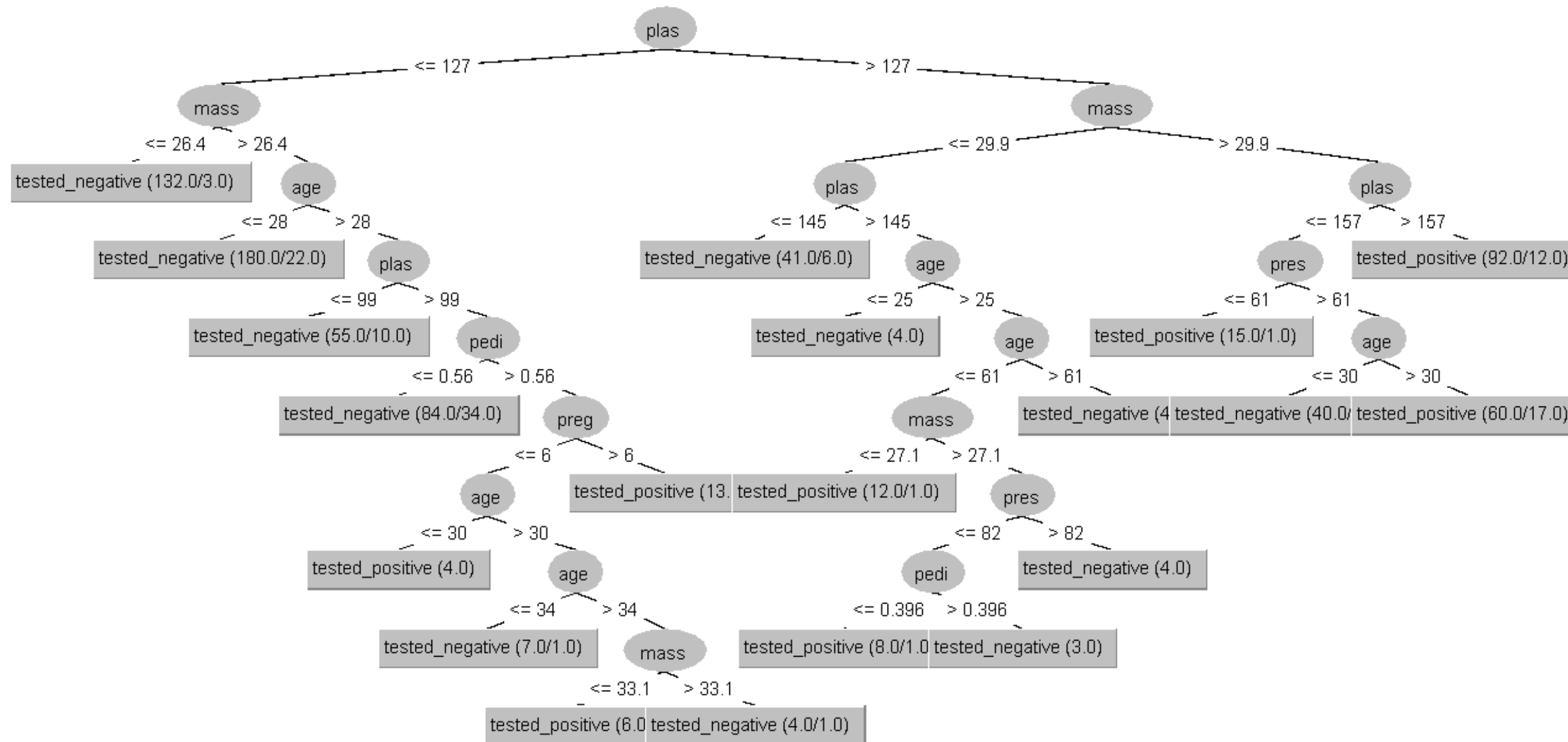
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.836	0.388	0.801	0.836	0.818	0.458	0.820	0.886	tested_negative
	0.612	0.164	0.667	0.612	0.638	0.458	0.820	0.679	tested_positive
Weighted Avg.	0.758	0.310	0.754	0.758	0.755	0.458	0.820	0.814	

=== Confusion Matrix ===

a	b	<-- classified as
418	82	a = tested_negative
104	164	b = tested_positive

2.9. Casos de Uso de Clasificación – Árbol de Decisión

Algoritmo: J48



- ✓ El nodo raíz y la variable mas importante es la variable **plas**
- ✓ Las otras variables mas importantes la **mass**, **age**, **pres**
- ✓ El árbol se lee de la siguiente forma:
 - si el plas es menor o igual a 127 y mass es menor o igual a 26.4 entonces es negativo a diabetes
 - Si el plas es mayor a 127 y mass es mayor a 29.9 y plas es mayor a 157 entp

Aprendizaje Supervisado - Regresión

Cuando usamos regresión, el resultado es un número. Es decir, el resultado de la técnica de machine learning que estemos usando será un valor numérico, dentro de un conjunto infinito de posibles resultados.

Aquí van algunos ejemplos de regresión:

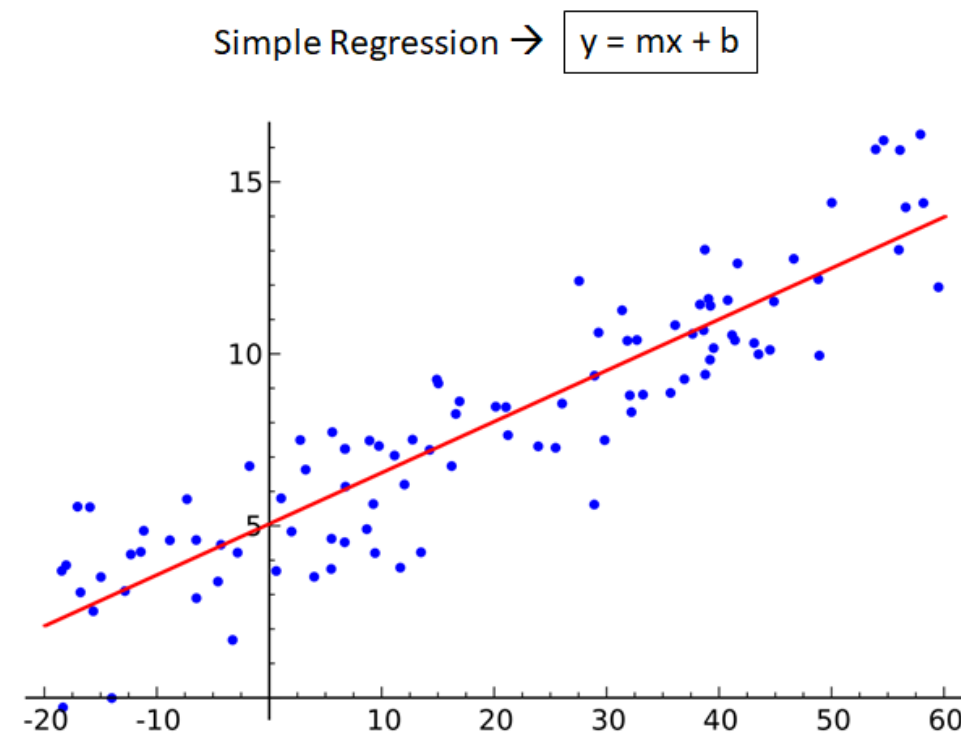
- ✓ Predecir por cuánto se va a vender una propiedad inmobiliaria
- ✓ Predecir cuantos accidentes de transito tendremos la siguiente semana
- ✓ Estimar cuantos estudiantes tendrán un mal desempeño escolar el próximo trimestre
- ✓ Estimar cuantos vehículos se matricularan en los próximos meses
- ✓ Estimar cuanto se va a vender en proxima época navideña

Aprendizaje Supervisado de Regresión

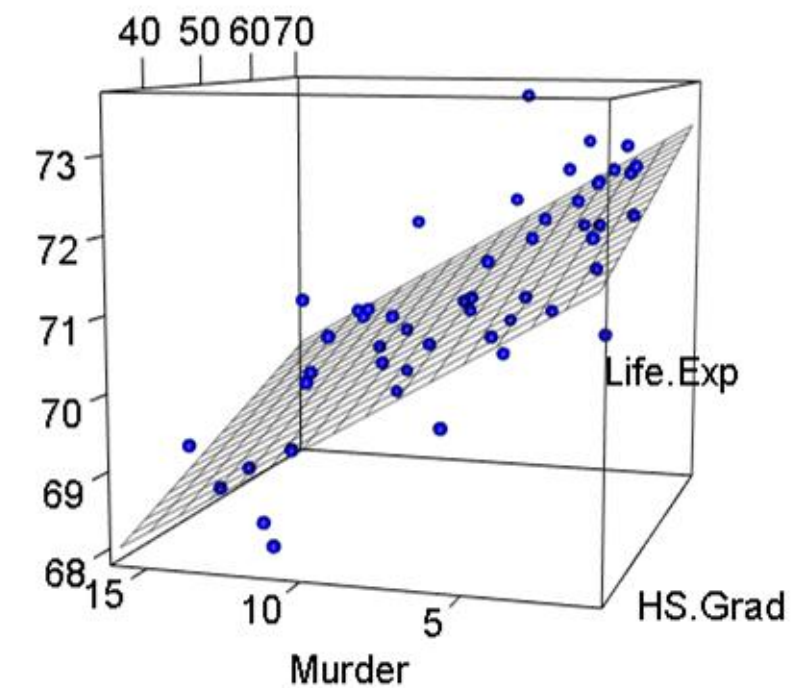
Un problema de regresión es cuando en base a los atributos del dataset, la variable objetivo o de salida es un valor real o continuo.

Se pueden usar muchos modelos diferentes, el más simple es la regresión lineal, el cual intenta ajustar los datos con el mejor hiperplano que pasa por los puntos.

Existe modelos de regresión simple y de regresión múltiple (mas de una variable atributo)



Multiple Regression → $y = m_1x_1 + m_2x_2 + b$

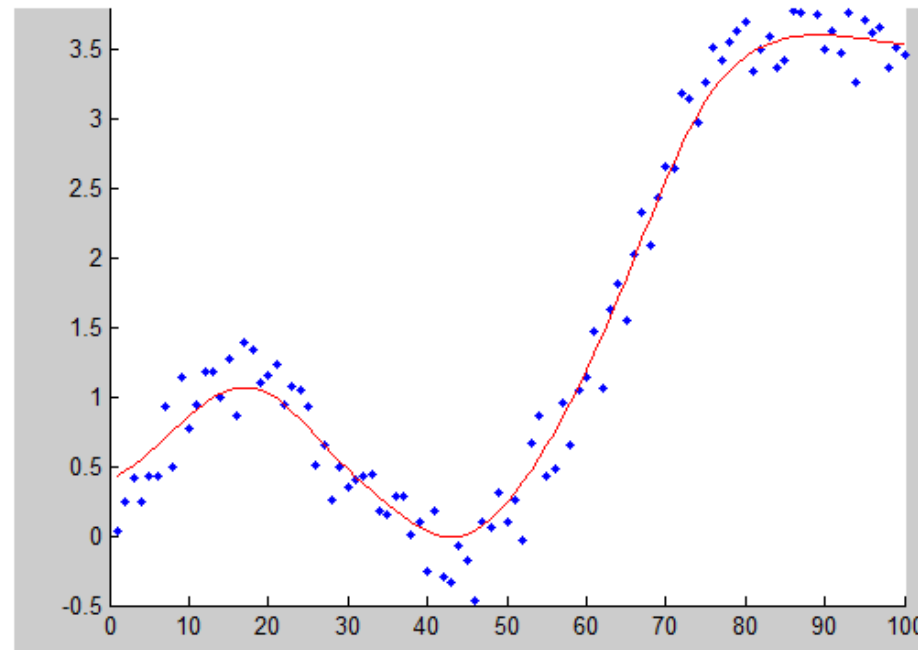


Aprendizaje Supervisado de Regresión

Además, los tipos de regresión son lineal y los regresión no lineal.

Polynomial Regression →

$$y = W_1x^3 + W_2x^2 + W_3x + W_4$$



Algunos ejemplos de aprendizaje supervisado de regresión:

- ✓ Predecir la edad de una persona
- ✓ Predecir las ventas del próximo mes de una empresa
- ✓ Predecir la temperatura el próximo domingo en la plaza mayor de Lima a las 4pm.

Aprendizaje Supervisado - Regresión

Hay varias técnicas de machine learning que podemos usar en problemas de Regresión. Podemos destacar:

- ✓ Regresión lineal y regresión no lineal
- ✓ **Arboles de decisión**
- ✓ Random forests
- ✓ Máquinas de soporte vectorial (support vector machines)
- ✓ Redes neuronales y aprendizaje profundo (deep learning)

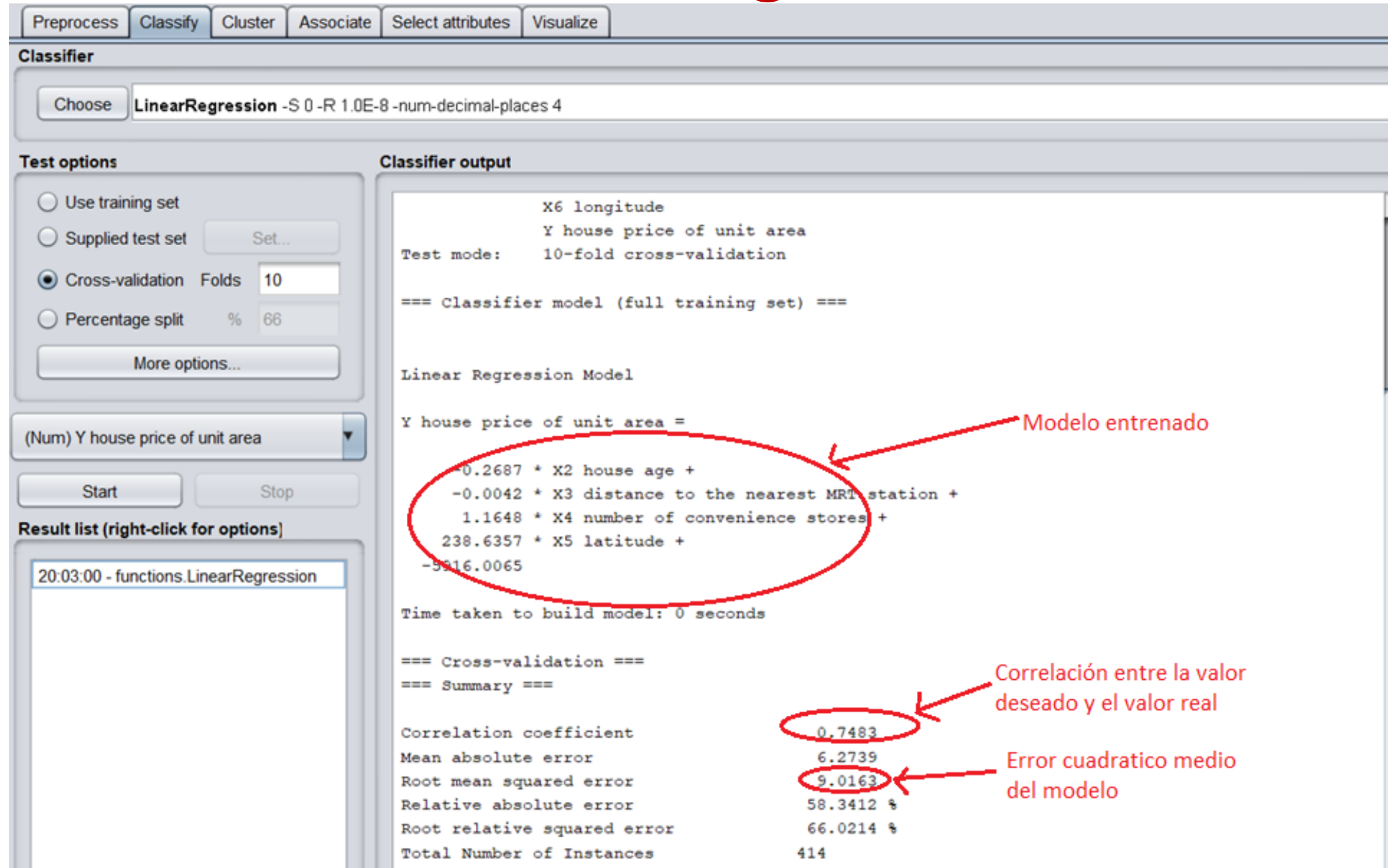
Regresión Lineal

En una Regresión Lineal, las variables independientes también se conocen como predictoras, que son las variables utilizadas para realizar predicciones sobre otras variables, a las que llamamos variables dependientes. Cuando la salida que perseguimos predecir depende de más de una variable, se puede utilizar un modelo más complejo que tenga en cuenta las dimensiones adicionales. Considerando si son relevantes o no para abordar el problema planteado, el uso de más variables puede contribuir a conseguir mejores predicciones.

La regresión lineal implica una serie de asunciones, y veremos que no es el mejor modelo para todas las situaciones.

- a) La regresión lineal funciona mejor con datos “lineales”, si no es así, será preciso realizar ajustes (transformar los datos de entrenamiento), añadir características, o usar otro modelo.
- b) La regresión lineal es sensible a los valores “extremos” de los datos, por lo que es preciso vigilar estos valores extremos y normalmente habrá que eliminarlos.

Regresión Lineal



Classifier

Choose **LinearRegression** -S 0 -R 1.0E-8 -num-decimal-places 4

Test options

- ☐ Use training set
- ☐ Supplied test set
- ☒ Cross-validation Folds
- ☐ Percentage split %
-

(Num) Y house price of unit area

Result list (right-click for options)

20:03:00 - functions.LinearRegression

Classifier output

```
X6 longitude
Y house price of unit area
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Linear Regression Model

Y house price of unit area =
-0.2687 * X2 house age +
-0.0042 * X3 distance to the nearest MRT station +
1.1648 * X4 number of convenience stores +
238.6357 * X5 latitude +
-5516.0065

Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient      0.7483
Mean absolute error         6.2739
Root mean squared error     9.0163
Relative absolute error     58.3412 %
Root relative squared error 66.0214 %
Total Number of Instances   414
```

Modelo entrenado

Correlación entre la valor deseado y el valor real

Error cuadrático medio del modelo

Resultados:

- ✓ Coeficiente de correlación: 0.7483
- ✓ Error cuadrático medio: 9.0163

Algoritmo de Regresión

Choose REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Num) Y house price of unit area

Start Stop

Result list (right-click for options)

20:35:48 - trees.REPTree

Classifier output

```
| X2 house age >= 11.2
| | X6 longitude < 121.54 : 38.89 (56/58.38) [30/44.18]
| | X6 longitude >= 121.54
| | | X3 distance to the nearest MRT station < 330.23 : 48.61 (30/77.31) [17/86.99]
| | | X3 distance to the nearest MRT station >= 330.23 : 39.32 (22/34.94) [18/34.91]
X3 distance to the nearest MRT station >= 763.37
| X5 latitude < 24.98
| | X3 distance to the nearest MRT station < 4007.27
| | | X2 house age < 17.75 : 27.12 (40/19.42) [19/12.81]
| | | X2 house age >= 17.75
| | | | X5 latitude < 24.95 : 17.26 (5/15.07) [3/18.18]
| | | | X5 latitude >= 24.95 : 24.79 (22/9.01) [6/24.7]
| | X3 distance to the nearest MRT station >= 4007.27 : 16.74 (18/10.65) [14/12.3]
| X5 latitude >= 24.98 : 36.81 (17/38.71) [10/122.69]
```

Size of the tree : 17

Time taken to build model: 0.01 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.7875
Mean absolute error	5.6204
Root mean squared error	8.438
Relative absolute error	52.2643 %
Root relative squared error	61.787 %
Total Number of Instances	414

Mide la correlación entre los resultados deseados y los obtenidos (mayor valor es mejor)

Es el error cuadrático medio de los valores deseados con los valores reales (menor valor es mejor)

Regression Example

- Algorithm: REPTree
- File: “Real estate valuation data set.csv”

The screenshot shows the WEKA software interface with the 'Classify' tab selected. The 'Classifier' dropdown is set to 'REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0'. Under 'Test options', 'Cross-validation' is selected with 'Folds' set to 10. The target variable is '(Num) Y house price of unit area'. The 'Start' button has been clicked, and the 'Classifier output' pane displays the following results:

```
| | | | X1 transaction date >= 2013.46 : 44.91 (8/54.84) [2/52.61]
| | X6 longitude >= 121.54
| | | X3 distance to the nearest MRT station < 330.23 : 48.61 (30/77.31) [17/86.99]
| | | X3 distance to the nearest MRT station >= 330.23 : 39.32 (22/34.94) [18/34.91]
X3 distance to the nearest MRT station >= 763.37
| X5 latitude < 24.98
| | X3 distance to the nearest MRT station < 4007.27
| | | X2 house age < 17.75 : 27.12 (40/19.42) [19/12.81]
| | | X2 house age >= 17.75
| | | | X5 latitude < 24.95 : 17.26 (5/15.07) [3/18.18]
| | | | X5 latitude >= 24.95 : 24.79 (22/9.01) [6/24.7]
| | X3 distance to the nearest MRT station >= 4007.27 : 16.74 (18/10.65) [14/12.3]
| X5 latitude >= 24.98 : 36.81 (17/38.71) [10/122.69]
```

Size of the tree : 31
Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient	0.7797
Mean absolute error	5.7935
Root mean squared error	8.607
Relative absolute error	53.8738 %
Root relative squared error	63.0238 %
Total Number of Instances	414

The 'Result list' pane shows a single entry: '18:50:50 - trees.REPTree'. The 'Status' bar at the bottom indicates 'OK'.

DataSet: Real Estate Valuation

Este archivo contiene información sobre el precio de ventas de casas en una localidad en base a algunos atributos.

Atributos:

- X2 house age: Antigüedad de las casas
- X3 distance to the nearest MRT station: Distancia a la estación de metro mas cercana
- X4 number of convenience stores: Número de tiendas de alimentos cerca a la vivienda
- X5 latitude: Latitud de la casa
- X6 longitude: Longitud de la casa

Variable Objetivo:

- Y house price of unit area: Precio de la casa por unidad de area

DataSet: Real Estate Valuation

Tab Preprocess -> Edit...

Viewer

Relation: Real estate valuation data set-weka.filters.unsupervised.attribute.Remove-R1-2

No.	1: X2 house age Numeric	2: X3 distance to the nearest MRT station Numeric	3: X4 number of convenience stores Numeric	4: X5 latitude Numeric	5: X6 longitude Numeric	6: Y house price of unit area Numeric
1	32.0	84.87882	10.0	24.98298	121.54024	37.9
2	19.5	306.5947	9.0	24.98034	121.53951	42.2
3	13.3	561.9845	5.0	24.98746	121.54391	47.3
4	13.3	561.9845	5.0	24.98746	121.54391	54.8
5	5.0	390.5684	5.0	24.97937	121.54245	43.1
6	7.1	2175.03	3.0	24.96305	121.51254	32.1
7	34.5	623.4731	7.0	24.97933	121.53642	40.3
8	20.3	287.6025	6.0	24.98042	121.54228	46.7
9	31.7	5512.038	1.0	24.95095	121.48458	18.8
10	17.9	1783.18	3.0	24.96731	121.51486	22.1
11	34.8	405.2134	1.0	24.97349	121.53372	41.4
12	6.3	90.45606	9.0	24.97433	121.5431	58.1
13	13.0	492.2313	5.0	24.96515	121.53737	39.3
14	20.4	2469.645	4.0	24.96108	121.51046	23.8
15	13.2	1164.838	4.0	24.99156	121.53406	34.3
16	35.7	579.2083	2.0	24.9824	121.54619	50.5
17	0.0	292.9978	6.0	24.97744	121.54458	70.1
18	17.7	350.8515	1.0	24.97544	121.53119	37.4
19	16.9	368.1363	8.0	24.9675	121.54451	42.3
20	1.5	23.38284	7.0	24.96772	121.54102	47.7
21	4.5	2275.877	3.0	24.96314	121.51151	29.3
22	10.5	279.1726	7.0	24.97528	121.54541	51.6
23	14.7	1360.139	1.0	24.95204	121.54842	24.6
24	10.1	279.1726	7.0	24.97528	121.54541	47.9
25	39.6	480.6977	4.0	24.97353	121.53885	38.8
26	29.3	1487.868	2.0	24.97542	121.51726	27.0
27	3.1	383.8624	5.0	24.98085	121.54391	56.2
28	10.4	276.449	5.0	24.95593	121.53913	33.6
29	19.2	557.478	4.0	24.97419	121.53797	47.0
30	7.1	451.2428	5.0	24.97562	121.54604	57.1

REPTree

Modelo de Arbol Entrenado:

- ✓ La Raiz del Arbol y variable mas importante para el modelo es **distancia a la estación de Tren mas cercano.**
- ✓ **Antigüedad de la casa y latitud y longitud** también son variables importantes en el modelo

