



Evaluación de varios modelos de clasificación

Calidad de vinos

El detalle de las columnas que tiene el dataset es:

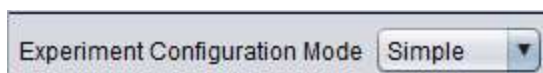
acidez fija	densidad
acidez volátil	PH
ácido cítrico	sulfatos
azúcar residual	alcohol
cloruros	calidad
libre dióxido de azufre	
dióxido de azufre total	

Se busca determinar, qué características son las que contribuyen a una mejor calidad en el vino blanco. La calidad se especifica a partir de 5 clases: **3, 4, 5, 6, 7 y 8**.

Una vez que ya conocemos los puntos importantes sobre la evaluación, vamos a contrastar más de un modelo de clasificación. Para lograr esto, vamos a utilizar el experimentador de Weka (Experimenter):



A partir de esta interfaz es que podemos evaluar más de un modelo de clasificación, con algunas configuraciones interesantes. Se tienen dos modos de experimentación, simple y avanzada



Experiment Configuration Mode: **Advanced** ▼

Open... Save... New

Destination

Choose

Result generator

Choose

Vamos a dar clic en la opción New, para poder iniciar un nuevo experimento de evaluación:

Weka Experiment Environment

Setup Run Analyse

Experiment Configuration Mode: **Simple** ▼

Open... Save... New

Results Destination

ARFF file ▼ Filename: Browse...

Experiment Type

Cross-validation ▼

Number of folds:

☒ Classification ☐ Regression

Iteration Control

Number of repetitions:

☒ Data sets first ☐ Algorithms first

Datasets

Add new... Edit selected... Delete selected

☐ Use relative paths

Up Down

Algorithms

Add new... Edit selected... Delete selected

Load options... Save options... Up Down

Notes

En esta parte, podemos determinar dónde guardar los resultados del análisis y el formato del archivo.

Tenemos que indicar el tipo de experimento que queremos realizar. Por omisión nos indica validación de cruzada a 10 folds. Las opciones que se tienen son:

Cross-validation ▼

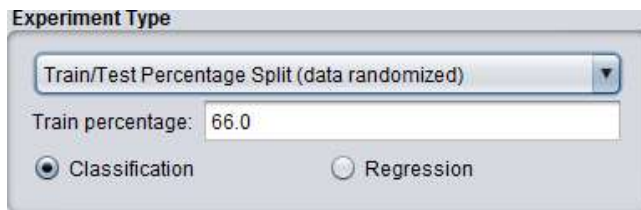
Cross-validation

Train/Test Percentage Split (data randomized)

Train/Test Percentage Split (order preserved)



Para nuestro primer experimento, utilizaremos retención (2/3 de tuplas para entrenamiento) y el 1/3 restante para prueba:



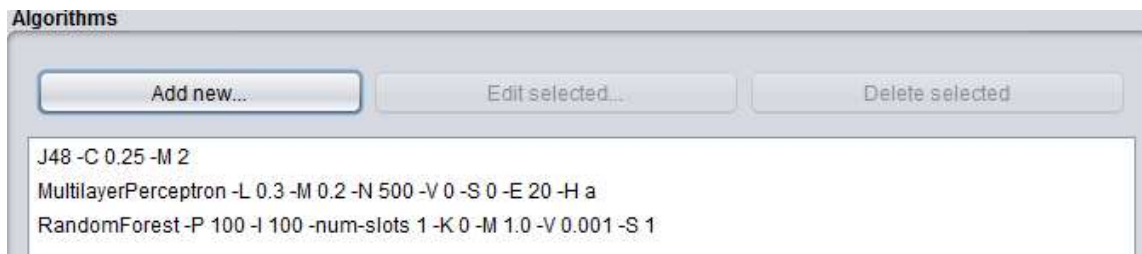
The 'Experiment Type' dialog box shows a dropdown menu set to 'Train/Test Percentage Split (data randomized)'. Below it, the 'Train percentage' is set to 66.0. At the bottom, the 'Classification' radio button is selected, and the 'Regression' radio button is unselected.

Vamos a agregar el dataset que evalua la calidad de vino blanco:



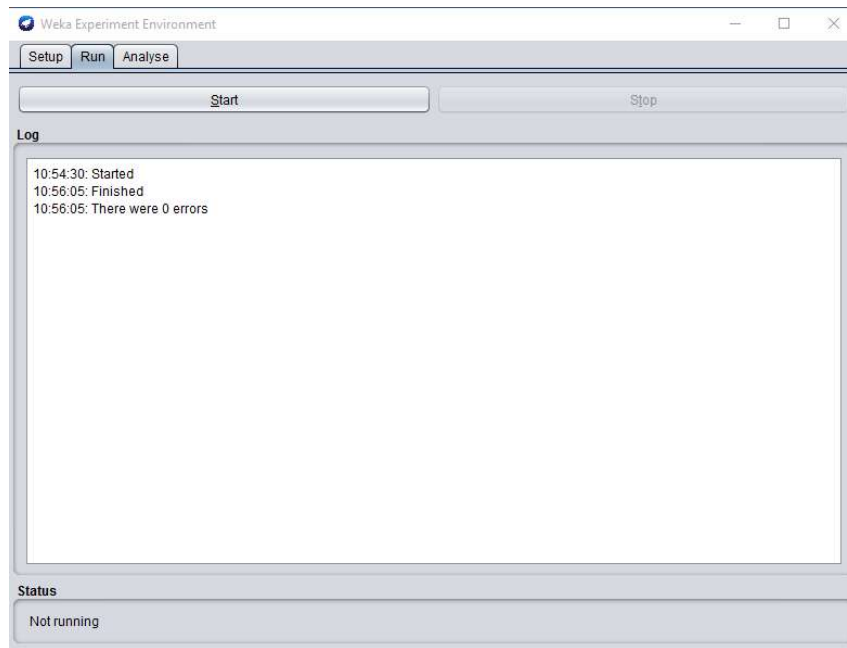
The 'Datasets' dialog box features buttons for 'Add new...', 'Edit selected...', and 'Delete selected'. A checkbox for 'Use relative paths' is present and unchecked. A text field contains the file path 'C:\Users\Gerardo\Desktop\Clasificacion\calidad_vino.csv'. At the bottom, there are 'Up' and 'Down' buttons.

Finalmente, vamos indicarlos modelos que deseamos comparar:

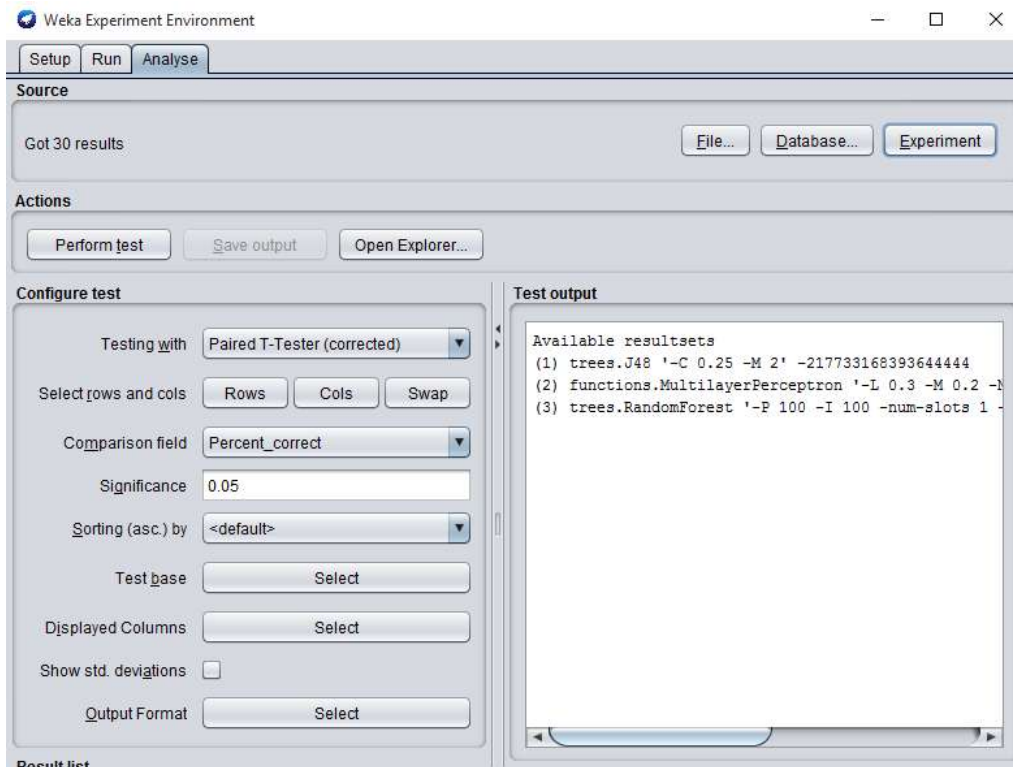


The 'Algorithms' dialog box includes buttons for 'Add new...', 'Edit selected...', and 'Delete selected'. The list of algorithms contains three entries: 'J48 -C 0.25 -M 2', 'MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a', and 'RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1'.

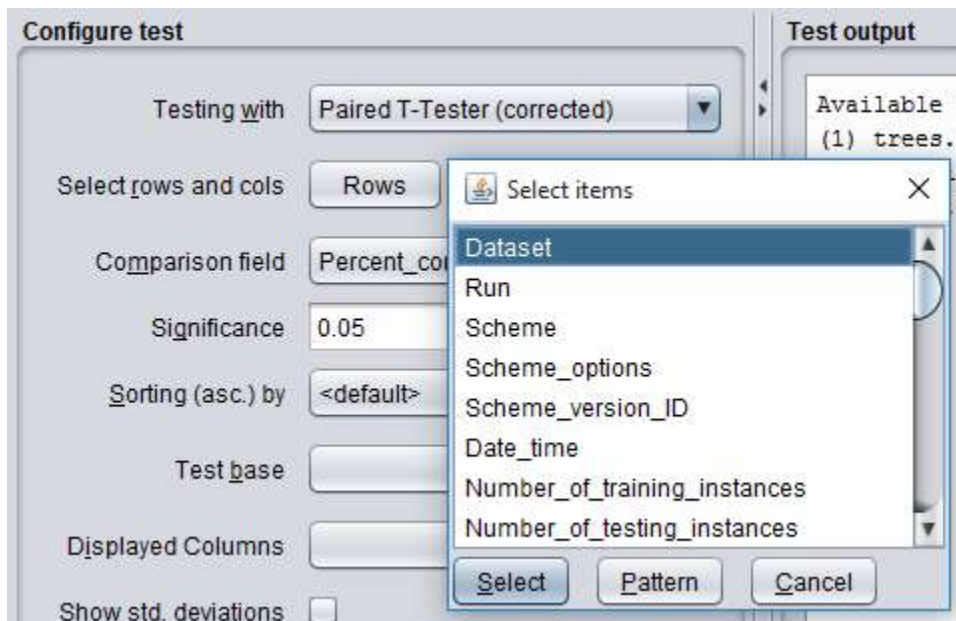
Nos movemos a la pestaña Run, para comenzar con el entrenamiento y prueba de los algoritmos seleccionados y esperamos a que terminen dichas fases:



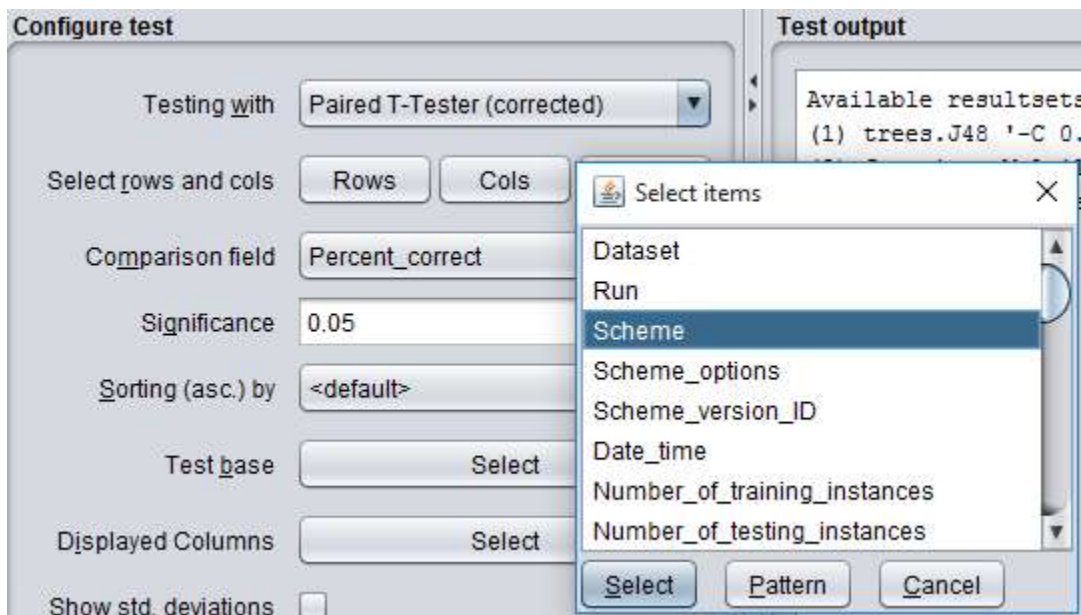
Cambiamos a la pestaña de Analyse y vamos a dar clic en Experiment:



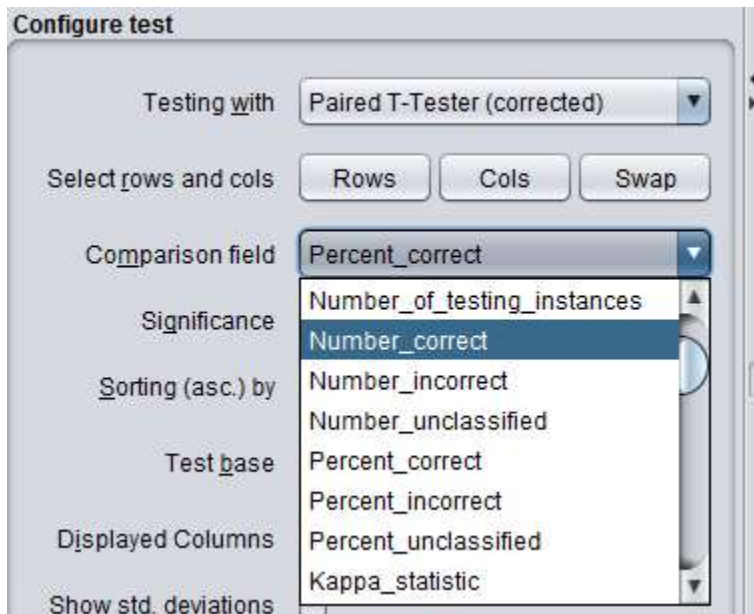
Vamos a realizar la siguiente configuración. En la parte de filas (**DATASET**):



En la parte de columnas (**SCHEME**):



En el campo de comparación, vamos a seleccionar la métrica con la cual deseamos realizar el experimento de contraste (**NUMBER_CORRECT**):



Para realizar la prueba, tenemos como opciones (**PAIRED T-TESTER (CORRECTED)**):

Una prueba T, asume que las muestras son independientes, pero si se aplica validación cruzada, las muestras no son independientes, si se ignora esta suposición, se generan muchos errores tipo 1 (se rechaza la hipótesis nula cuando era cierta). La prueba T pareada corregida aplica un factor que permite contrarrestar la dependencia entre las muestras que en la práctica resulta en errores aceptables de tipo I.



Una **prueba t** es una **prueba de hipótesis de la media de una o dos poblaciones distribuidas normalmente**. Aunque existen varios tipos de prueba t para situaciones diferentes, en todas se utiliza un estadístico de prueba que sigue una distribución t bajo la hipótesis nula:

Prueba	Propósito	Ejemplo
Prueba t de 1 muestra	Prueba si la media de una población individual es igual a un valor objetivo	¿Es la estatura media de las estudiantes universitarias mayor que 5.5 pies?
Prueba t de 2 muestras	Prueba si la diferencia entre las medias de dos poblaciones independientes es igual a un valor objetivo	¿Difiere significativamente la estatura media de las estudiantes universitarias con respecto a la de los estudiantes universitarios?
Prueba t pareada	Prueba si la media de las diferencias entre las observaciones dependientes	Si usted registra el peso de estudiantes universitarios antes y después de que cada uno de ellos tome una píldora para adelgazar, ¿es suficientemente significativa la

Prueba	Propósito	Ejemplo
	o pareadas es igual a un valor objetivo	pérdida media de peso para llegar a la conclusión de que la píldora es efectiva?
Prueba t en la salida de regresión	Prueba si los valores de los coeficientes en la ecuación de regresión difieren significativamente de cero	¿Son predictores significativos de los GPA universitarios las puntuaciones de las pruebas SAT de educación secundaria?

Una **propiedad importante de la prueba t** es su **robustez ante los supuestos de normalidad de la población**. En otras palabras, las pruebas t suelen ser válidas incluso cuando se viola el supuesto de normalidad, pero solo si la distribución no es muy asimétrica. Esta propiedad la convierte en uno de los procedimientos más útiles para hacer inferencias sobre las medias de las poblaciones.

Sin embargo, con distribuciones no normales y muy asimétricas, podría ser más conveniente usar pruebas no paramétricas.

Para la prueba t, las hipótesis son:

Hipótesis nula

$$H_0: \mu_d = \mu_0$$

La media de las diferencias de la población (μ_d) es igual a la media hipotética de las diferencias (μ_0).

Hipótesis alternativa

Elija una:

$$H_1: \mu_d \neq \mu_0$$

La media de las diferencias de la población (μ_d) no es igual a la media hipotética de las diferencias (μ_0).

$$H_1: \mu_d > \mu_0$$

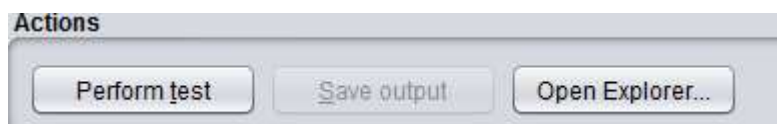
La media de las diferencias de la población (μ_d) es mayor que la media hipotética de las diferencias (μ_0).

$$H_1: \mu_d < \mu_0$$

La media de las diferencias de la población (μ_d) es menor que la media hipotética de las diferencias (μ_0).

- **Pareado** si las dos medias son tomadas de la misma muestra estadística: Comparación con las mismas muestras procesadas en dos métodos.
- **No pareados** cuando se comparan dos medias de diferente muestra estadística

Una vez configurado, vamos a efectuar la prueba:



Test output

```

Tester:      weka.experiment.PairedCorrectedTTester -G 3 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText -mean-p
Analysing:   Percent_correct
Datasets:    1
Resultsets:  3
Confidence:  0.05 (two tailed)
Sorted by:   -
Date:        23/05/17 11:22 AM

```

Dataset	(1) trees.J4	(2) funct	(3) trees
calidad_vino	(10) 58.57	58.76	67.52 v
	(v/ /*)	(0/1/0)	(1/0/0)

Key:

- (1) trees.J48
- (2) functions.MultilayerPerceptron
- (3) trees.RandomForest

Vamos a analizar los resultados:

```

Tester:      weka.experiment.PairedCorrectedTTester -G 3 -D 1 -F
Analysing:   Percent_correct
Datasets:    1
Resultsets:  3
Confidence:  0.05 (two tailed)
Sorted by:   -
Date:        23/05/17 11:22 AM

```

Dataset	(1) trees.J4	(2) funct	(3) trees
calidad_vino	(10) 58.57	58.76	67.52 v
	(v/ /*)	(0/1/0)	(1/0/0)

Key:

- (1) trees.J48
- (2) functions.MultilayerPerceptron
- (3) trees.RandomForest

```

Tester:      weka.experiment.PairedCorrectedTTester -G 3 -D 1 -R 2 -!
Analysing:   Percent_correct
Datasets:    1
Resultsets:  3
Confidence:  0.05 (two tailed)
Sorted by:   -
Date:        23/05/17 11:34 AM

```

```

a      b      c (No. of datasets where [col] >> [row])
- 1 (0) 1 (1) | a = (1) trees.J48
0 (0)      - 1 (1) | b = (2) functions.MultilayerPerceptron
0 (0) 0 (0)      - | c = (3) trees.RandomForest

```

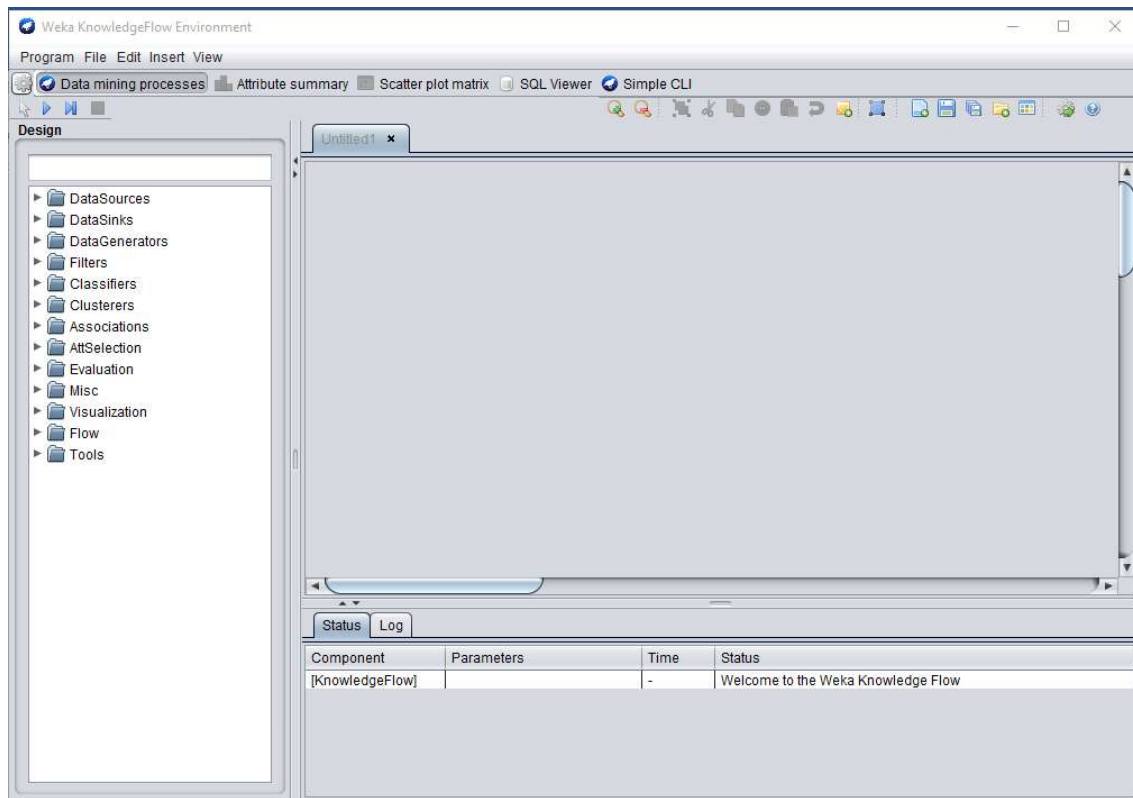



```
Tester:      weka.experiment.PairedCorrectedTTester -G
Analysing:   Percent_correct
Datasets:    1
Resultsets:  3
Confidence:  0.05 (two tailed)
Sorted by:   -
Date:        23/05/17 11:39 AM
```

```
>-<  >  < Resultset
      2  2  0 trees.RandomForest
     -1  0  1 functions.MultilayerPerceptron
     -1  0  1 trees.J48
```

Finalmente, podemos comparar los modelos para ver la curva ROC con el flujo de conocimiento de Weka:





Vamos a configurar el siguiente experimento a través de nodos en un flujo, los objetos que se van a seleccionar son:

- **CSVLOADER: uno**
- **CLASSASSIGNER: uno**
- **CLASSVALUEPICKER: uno**
- **CROSSVALIDATIONFOLDMAKER: uno**
- **J48: uno**
- **MULTILAYERPERCEPTRON: uno**
- **CLASSIFIERPERFORMANCEEVALUATOR: dos**
- **MODELPERFORMANCECHART: uno**
- **COSTBENEFITANALISYS: uno**

