



Escuela Politécnica
Superior (EPS)



UNIVERSIDAD DE CÓRDOBA
Departamento de
Informática y
Análisis Numérico

Práctica 3 - Regresión y clasificación con 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

Índice de contenidos

Regresión

Regresión Lineal

Clasificación

K-NN

Regresión Logística

Entregables

Bibliografía



Regresión

Regresión Lineal

Clasificación

K-NN

Regresión Logística

Entregables

Bibliografía

¿Qué es Regresión?

- Consiste en la estimación de una **variable dependiente** que toma valores reales, a partir de una serie de **N** variables independientes, es decir:
 - ▶ Dado: $(x_1, y_1), \dots, (x_m, y_m)$, siendo **m** el número de patrones etiquetados (se conoce el valor de la variable dependiente - experto), encontrar una función f tal que
$$f : X_1 x X_2 x \dots x X_N \longrightarrow Y, x_j \in 1 \leq j \leq N$$
 - ▶ Es decir, buscamos un $f(x)$ que sea un buen predictor para la variable Y , utilizando para ello una **función de error (métrica)**.
 - ▶ Por ejemplo mediante una **función lineal o recta en el plano**:
$$Y = \beta_0 + \beta_1 x + \varepsilon \quad \varepsilon \sim N(0, \sigma),$$
donde ε es una variable aleatoria que representa información predictora que no se haya tenido en cuenta.
 - ▶ $\beta_0, \beta_1, \beta_N$ miden la influencia que las variables independientes sobre la dependiente.
 - El **ajuste de la función lineal** se evaluará teniendo en cuenta el **error cometido**. Ej: Error como **sumatorio para cada patrón de lo mala que ha sido una predicción**.

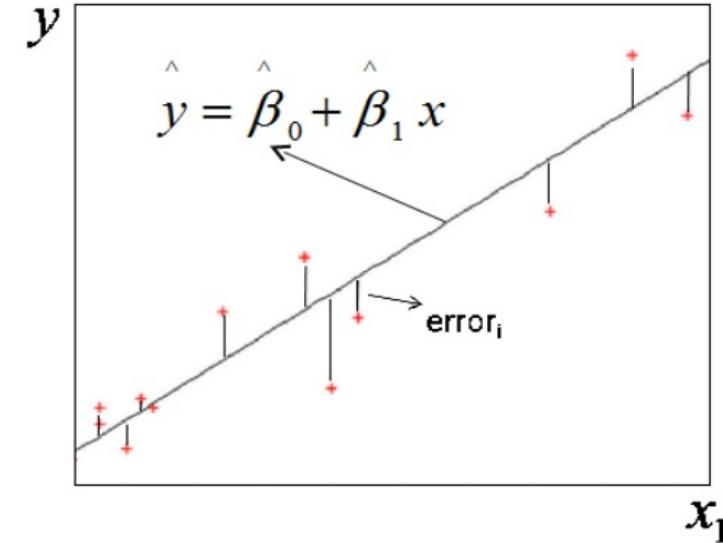
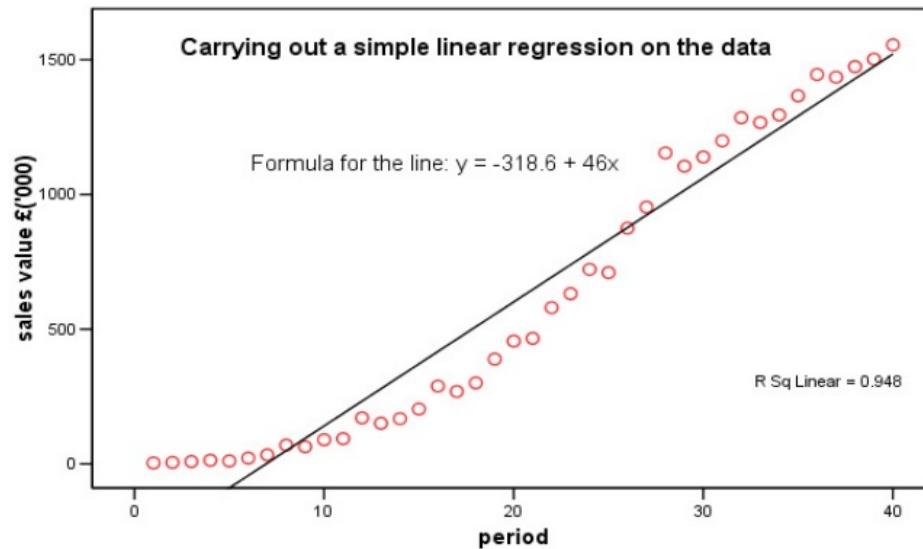
Ajuste del error en regresión

Objetivo

- Minimizar la **suma de errores cuadráticos** (*squared sum of errors - SSE*):

$$SSE = \sum_{i=1}^m (y_i - \hat{y}_i)^2 = \sum_{i=1}^m \left(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) \right)^2$$

- y_i : Valor real para el patrón i de un total de m patrones.
 - \hat{y}_i : Valor predicho para el patrón i de un total de m patrones. $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$



Métricas de error en regresión

- p_i : Valor predicho para el patrón i de un total de m patrones.
 - r_i : Valor real para el patrón i de un total de m patrones.
 - \bar{r} : Media de los valores reales.

Medidas de Regresión en Weka (a maximizar sobre el conjunto de test)

- *Correlation coefficient (R^2):* Valor entre [0, 1].

$$R^2 = 1 - \frac{\sum_{i=1}^m (p_i - r_i)^2}{\sum_{i=1}^m (r_i - \bar{r})^2}$$

- Se basa en el coeficiente de correlación de Pearson.
 - Mide qué ratio de la varianza de Y es explicada por el modelo (cómo de bien se ajustan los datos a la recta de regresión).

Métricas de error en regresión

Medidas de Regresión en Weka (a minimizar sobre el conjunto de test)

- *Mean Absolute Error (MAE)* (0- ∞): Mide el promedio de los errores cometidos en un conjunto de predicciones.

$$MAE = \frac{|p_1 - r_1| + \dots + |p_m - r_m|}{m}$$

- *Root Mean Squared Error (RMSE)* (0- ∞): Mide el promedio cuadrático de los errores cometidos en un conjunto de predicciones. Los errores grandes aumentan mucho el valor de esta métrica, es sensible a atípicos, pero nos permite “intuir” su existencia.

$$RMSE = \sqrt{\frac{(p_1 - r_1)^2 + \dots + (p_m - r_m)^2}{m}}$$

Métricas de error en regresión

Medidas de Regresión en Weka (a minimizar sobre el conjunto de test)

- *Relative Absolute Error (RAE)* (0-∞) en %: Error con respecto al error que se cometería al predecir la media (**regresor trivial**).

$$RAE = \frac{|p_1 - r_1| + \dots + |p_m - r_m|}{|\bar{r} - r_1| + \dots + |\bar{r} - r_m|}$$

Ej: Si se obtiene un 1 (100 %) quiere decir que se tiene el mismo error que un modelo que predice la media.

Ej: Si se obtiene un 0.5 (50 %) sería la mitad del error que produce un modelo trivial que predeciría la media.

Ej: Más de un 100 % significa que el predictor es peor que un predictor trivial.

- *Root Relative Squared Error (RRSE)* (0-∞) en %: Igual que el RAE, pero al ser cuadrático exageran los errores más grandes mientras que dan menos importancia a los errores pequeños.

$$RRSE = \sqrt{\frac{(p_1 - r_1)^2 + \dots + (p_m - r_m)^2}{(\bar{r} - r_1)^2 + \dots + (\bar{r} - r_m)^2}}$$

Regresión
oooooo

Regresión Lineal
●oooooooooooo

Clasificación
oooooooooooooooooooo

k-NN
ooo

Regresión Logística
oooooooooooo

Entregables
oooo

Bibliografía
o

Regresión

Regresión Lineal

Clasificación

K-NN

Regresión Logística

Entregables

Bibliografía

Regresión Lineal Simple y Regresión Lineal Multiple

Regresión Lineal

- Las variable **dependiente** es **numérica** (**no es clasificación**).
- Se asume una **relación lineal** (línea recta) entre las variables independientes y la dependiente.
$$Y = \beta_0 + \beta_1 x + \varepsilon \quad \varepsilon \sim N(0, \sigma)$$
- Relación de las β respecto a la variable.
 - ▶ $\beta > 0 \Rightarrow$ Asociación positiva. Si aumenta la independiente aumenta la dependiente.
 - ▶ $\beta < 0 \Rightarrow$ Asociación negativa. Si aumenta la independiente decremente la dependiente.
 - ▶ $\beta = 0 \Rightarrow$ No existe asociación.

Regresión Lineal Simple y Regresión Lineal Multiple

- **RL. Simple:** Una **única** variable independiente. $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \varepsilon$
- **RL. Múltiple:** Existen **N** variables independientes.
$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \cdots + \hat{\beta}_n x_{ni} + \varepsilon$$

Regresión Lineal en Weka

- **Regresión Lineal Simple:**

classifiers/functions/SimpleLinearRegression

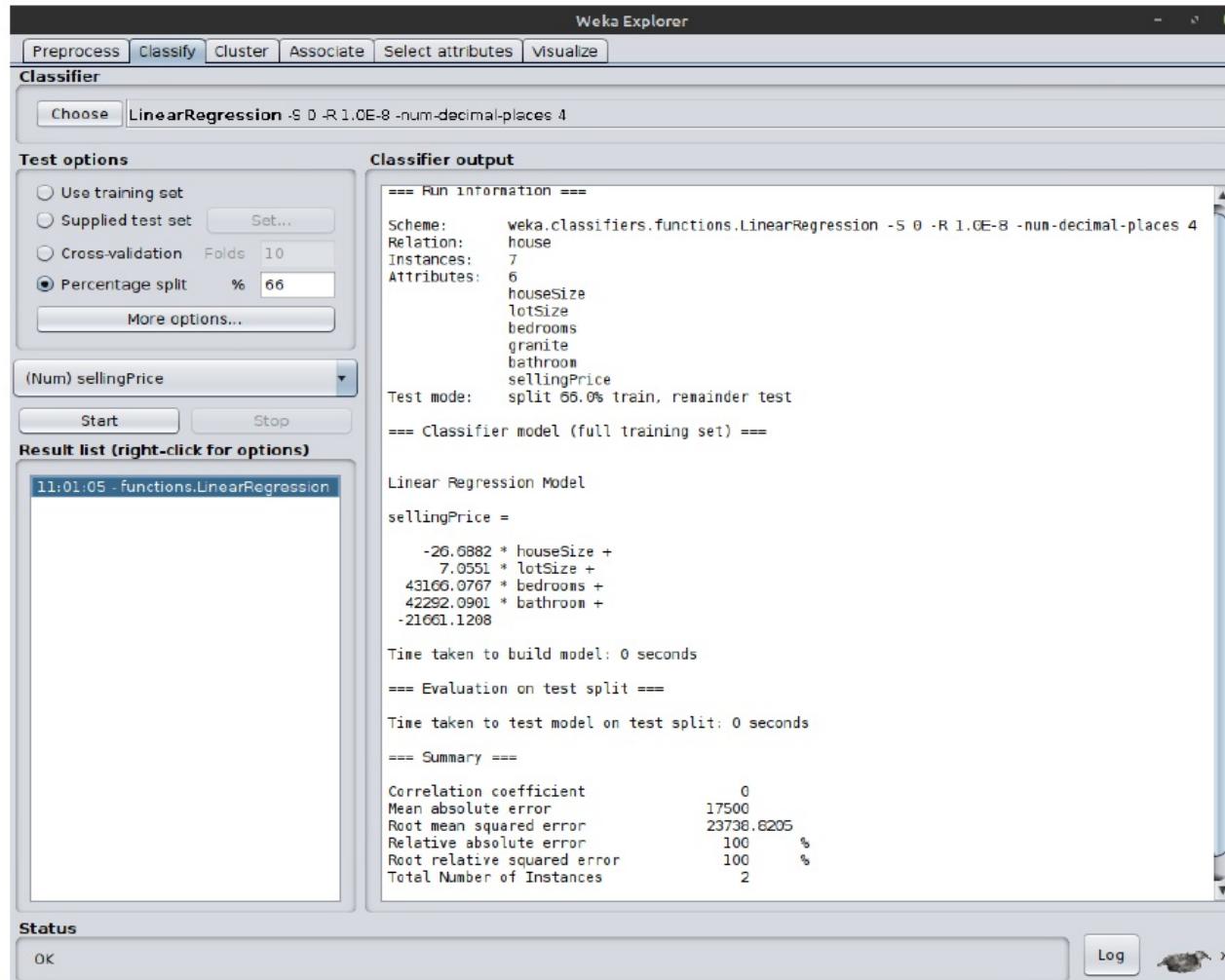
- ▶ *classifiers/functions/SimpleLinearRegression* solo trabaja con **atributos numéricos**.
- ▶ Necesidad de filtros de preprocessado ante atributos nominales. *filters/supervised/attribute/NominalToBinary*.

- **Regresión Lineal Múltiple:**

classifiers/functions/LinearRegression

Regresión Lineal en Weka

Estudio de la base de datos **house.arff** (disponible en Moodle).
Use el algoritmo ***LinearRegression*** de Weka para sacar conclusiones.

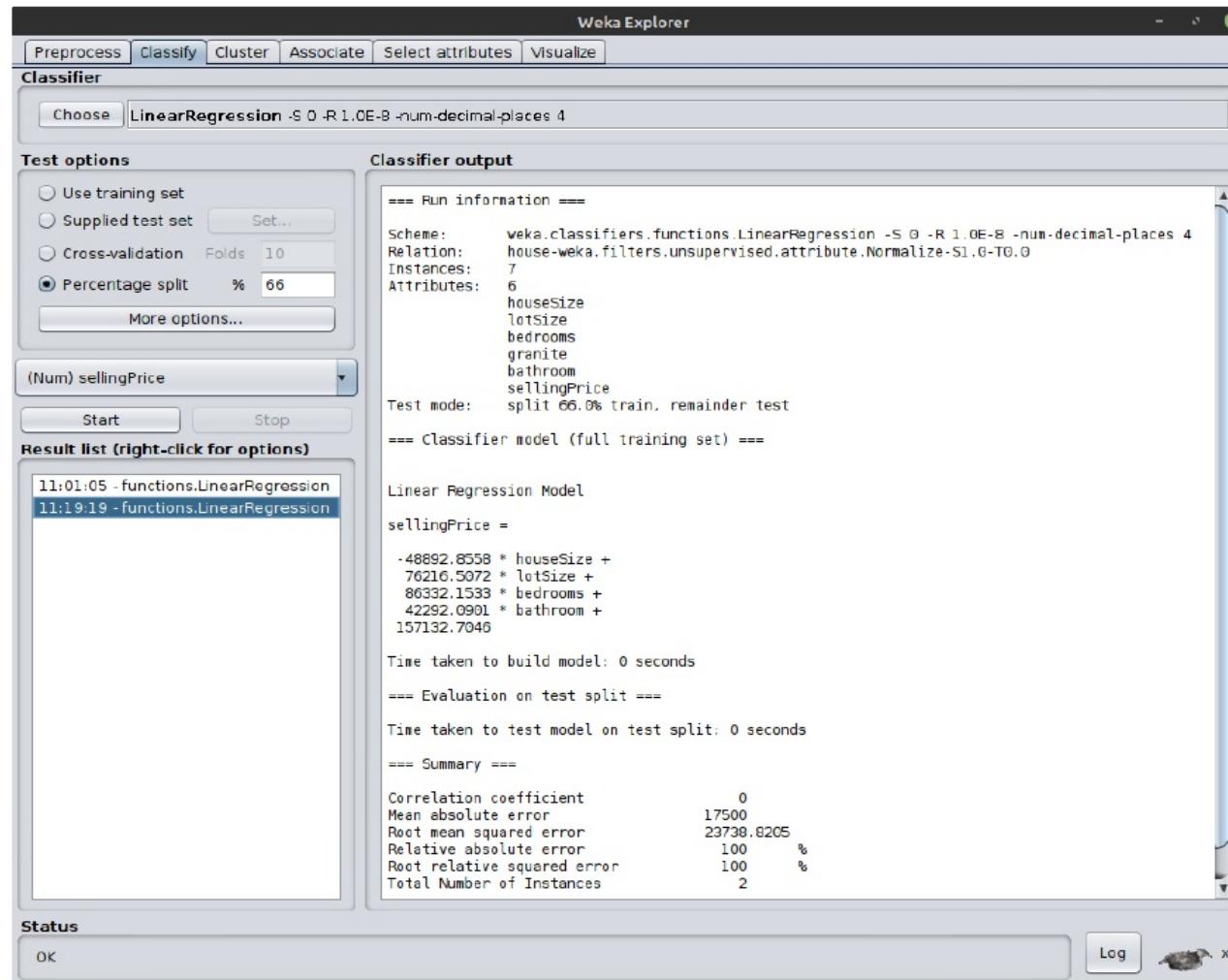


Regresión Lineal en Weka

- ¿Cuál será la variable dependiente si queremos predecir precios de venta? Recuerde que puede indicarla en las opciones de configuración antes de lanzar el experimento en la pestaña “*Classify*”.
 - “*sellingPrice*”, que es el precio de venta de una casa.
- ¿Qué ocurre con la variable *Granite*?
 - La variable “*granite*” no aparece en el modelo de regresión lineal que obtiene Weka, por tanto no es una variable independiente que aporte información sobre la dependiente. No nos sirve para el modelo.
- *Bathroom* es una variable numérica con valores 0 y 1. ¿Cómo influye en el modelo?
 - Con el valor 42292,0901 sería la segunda variable más importante, pero ¿hemos normalizado?

Regresión Lineal en Weka

Modelo de regresión lineal una vez usado el filtro filters/unsupervised/attribute/Normalize.

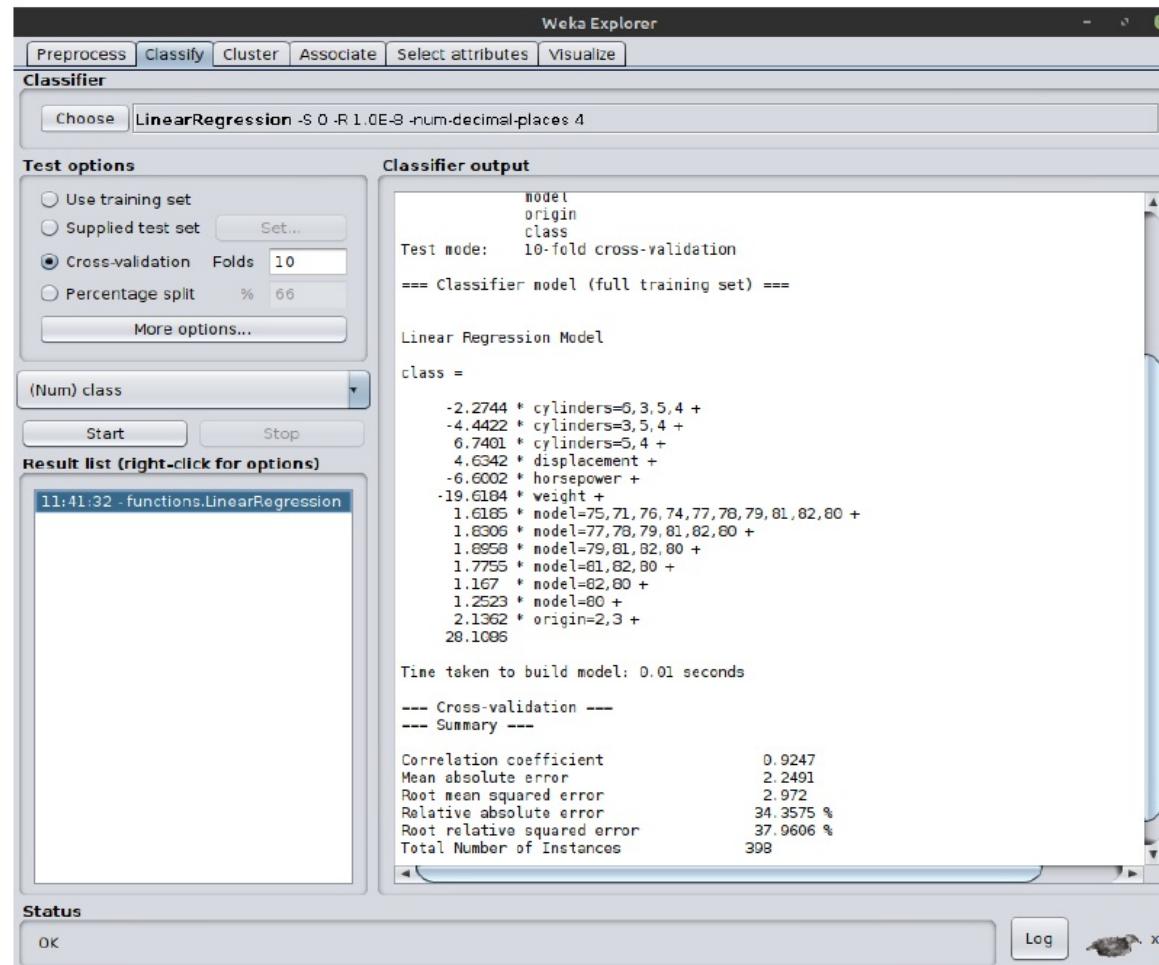


Regresión Lineal en Weka

- Ahora vemos que “Bathroom” ya no es tan importante, dentro de las variables que influyen sobre la salida es la menos importante.
- ¿Cómo influye el número de dormitorios (*bedrooms*)?
- Es la variable más importante respecto al valor de venta (mayor valor de β).
- ¿Qué podría decir de la variable *houseSize*?
- Parece que el tamaño de la casa influye negativamente sobre la salida, es decir, que a mayor tamaño el precio de venta es menor. Es una incoherencia o podría ser que se le da más importancia al resto de variables del modelo a la hora de vender.
- ¿Qué podría decir de las métricas de salida?
- Es un modelo bastante malo si observamos tanto R^2 como RAE y $RRSE$ (es un regresor trivial) y las variables de entrada no explican la salida del modelo.

Regresión Lineal en Weka

Estudio de la base de datos **autoMpg.arff** (disponible en Moodle).
Uso del algoritmo *LinearRegression* con un *10-fold* (normalice previamente).



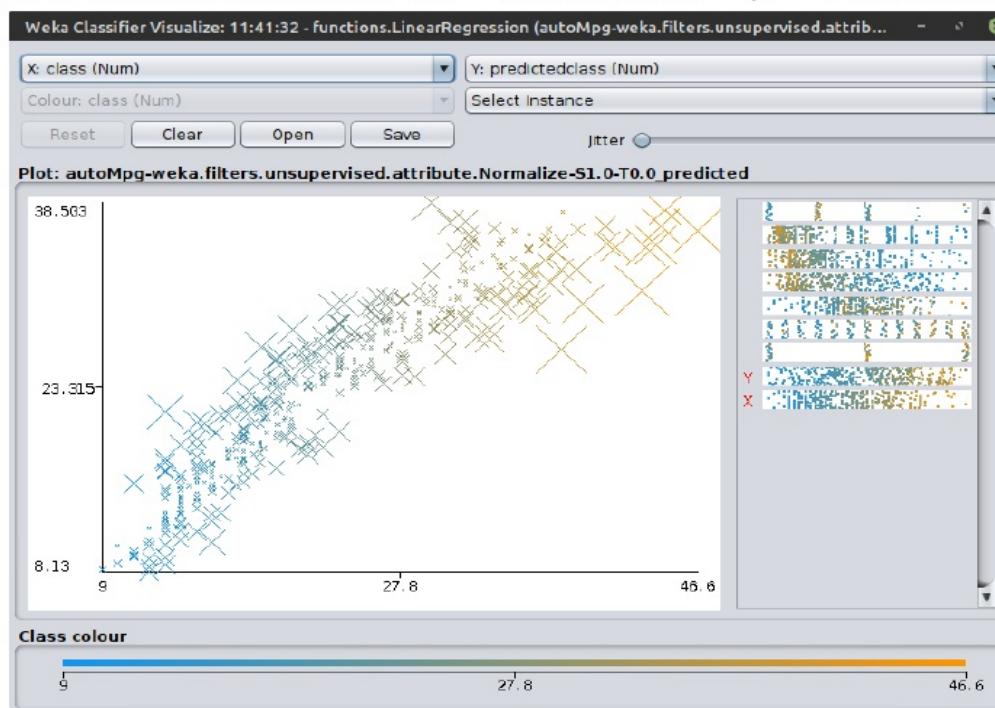
Regresión Lineal en Weka

autoMpg.arff es una base de datos que intenta predecir cuántas millas (*mpg*) recorrerá un vehículo con un galón de gasolina a partir de 7 características.

- ¿Cuál es el atributo que nos aporta mayor información de la variable dependiente?
 - El atributo "weight" con un valor de $-19,6184$. Tiene sentido, a mayor peso del vehículo menos millas recorrerá con un galón de gasolina.
- ¿Existe algún atributo que no aporte información al modelo?
 - El atributo "acceleration" no influye sobre la salida, no aparece (β próximo a cero).

Regresión Lineal en Weka

- Visualiza gráficamente los errores cometidos. ¿Qué representan las diferentes cruces?
- Botón derecho ratón sobre la salida en "Result list"->"Visualize Classifier Errors".



- En el eje X se sitúa el valor esperado de la variable dependiente, mientras que en el eje Y se sitúa el estimado.
- A mayor tamaño de las equis mayor error cometido. Hacer "click" sobre una equis grande y observad la ventana que aparece.

Regresión Lineal en Weka

→ ¿Cómo se interpreta el modelo con los valores nominales?

```
class =  
  
    -2.2744 * cylinders=6,3,5,4 +  
    -4.4422 * cylinders=3,5,4 +  
    6.7401 * cylinders=5,4 +  
    4.6342 * displacement +  
    -6.6002 * horsepower +  
    -19.6184 * weight +  
    1.6185 * model=75,71,76,74,77,78,79,81,82,80 +  
    1.8306 * model=77,78,79,81,82,80 +  
    1.8958 * model=79,81,82,80 +  
    1.7755 * model=81,82,80 +  
    1.167 * model=82,80 +  
    1.2523 * model=80 +  
    2.1362 * origin=2,3 +  
    28.1086
```

Vamos a suponer que nos llega un patrón no visto con estos valores en los atributos, el modelo de regresión anterior se quedaría así:

```
cylinders = 6  
displacement = 199  
horsepower= 90  
weight = 2000  
model = 70  
origin = 2
```

$y = 28.1086 - 2.2744*1 + 4.6342*199 - 6.6002*90 - 19.6184*2000 + 2.1362*2$

NOMINALES:

A los nominales que aparecen en el modelo se sustituye por el valor 1.

Los nominales que no aparecen no se tienen en cuenta (caso de model 70 en el ejemplo).

Si aparece más de una vez, se pone 1 en cada linea donde aparezca y se usa el beta asociado, con lo que se podría hacer la suma. Por ejemplo cylinders = 3 sería:

$-2.2744*1 -4.4422*1$

Regresión Lineal en Weka

→ ¿Qué podría decir de las métricas de salida?

- Es un modelo bueno si observamos tanto R^2 que está cercano a 1 como RAE y $RRSE$ (casi un 40 % mejor que un regresor que prediga siempre la media). $RMSE$ tiene un valor que no es demasiado grande si miramos la escala de la variable de salida, 2,972 millas de error medio.
- MAE y $RMSE$ son métricas que nos aportarán más a la hora de comparar rendimientos entre varios algoritmos, por si solas a veces son difíciles de interpretar.

Regresión
oooooo

Regresión Lineal
oooooooooooo

Clasificación
●oooooooooooo

k-NN
ooo

Regresión Logística
oooooooooooo

Entregables
oooo

Bibliografía
o

Regresión

Regresión Lineal

Clasificación

K-NN

Regresión Logística

Entregables

Bibliografía

Clasificación y sus métricas

En clasificación la variable de salida a predecir (dependiente) es un conjunto de etiquetas, una por cada clase.

¿Para qué sirven las métricas de evaluación?

- Miden el **rendimiento y calidad** de un modelo, su **error** cometido.
 - ▶ Existen multitud de métricas de evaluación en clasificación.
 - ▶ Un buen valor en una métrica no significa necesariamente buenos valores en las demás (al igual que en regresión).
- Se pueden emplear para **comparar el rendimiento** de varios modelos obtenidos mediante diferentes procedimientos.
- En **clasificación supervisada** la mayoría de las métricas surgen de lo que se llama **matriz de confusión**.

Matriz de confusión en Clasificación

¿Qué es una matriz de confusión?

- Es una tabla de errores que permite la visualización del desempeño de un **modelo supervisado**.
- Se obtiene a partir del **conjunto de generalización o testing** aplicado al modelo supervisado construido (sobre el **conjunto de training**).
- Cada **fila** representa a las instancias en la **clase real**.
- Cada **columna** representa la **clase inferida o predicha** por el modelo, que puede ser igual o no a la real.

| | | Predicción | |
|-------------------|----------------------|---------------------------|---------------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

Precisión Global o CCR

Accuracy o Correct Classification Rate (CCR)

| | | Predicción | |
|------------|-------|---------------------------|---------------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

- Muestra el porcentaje de patrones correctamente clasificados.
- Se suele expresar en tanto por ciento o tanto por uno (ej: 95 % o 0.95).
- Sirve para problemas **bi-clase** o **multi-clase**.

$$CCR = \frac{TP + TN}{TP + TN + FP + FN}$$

- En un problema **multiclasa**, la precisión global se calcula como la suma de los **elementos de la diagonal** de la matriz de confusión, dividido por la suma de todos los elementos de la matriz.

$$CCR = \frac{1}{N} \sum_{j=1}^J n_{jj},$$

donde N es el número de patrones en generalización, J es el número de clases, y n_{jj} (elemento de la diagonal) es el número de patrones de la clase $j-th$ que están correctamente clasificados.

Precisión Global o CCR

Problemática de la Precisión Global o CCR:

- **No** proporciona un **valor fiable** en bases de datos **desbalanceadas**.
- **Desbalanceo**: Problemas con clases distribuidas de manera no uniforme, muchos patrones de una o varias clases y pocos patrones de una o varias clases.
- Suponga un problema con 2 clases:
 - ▶ 9990 ejemplos de la clase 1
 - ▶ 10 ejemplos de la clase 2
- Si el modelo siempre dice que los ejemplos son de la clase 1, su precisión global es:

$$CCR = \frac{9990}{10000} = 99,9\%$$

- Valor engañoso, ya que **nunca detecta patrones de la clase 2**.
- La optimalidad de un clasificador podría depender de la distribución de clases (**balanceada o desbalanceada**) y de su capacidad de predicción en cada una de ellas.

Métricas para Clasificación Bi-clase

TP Rate, TN Rate, Precision, FP Rate y F-Measure

| | | Predicción | |
|------------|-------|---------------------------|---------------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

- **TP Rate, Recall, Sensitivity, Precisión positiva (A Maximizar):**
Porcentaje de patrones positivos predichos como positivos.

$$TPRate = Recall = \frac{TP}{TP+FN}$$

- **TN Rate, Specificity, Precisión negativa (A Maximizar):**
Porcentaje de patrones negativos predichos como negativos.

$$Specificity = \frac{TN}{FP+TN}$$

- **Precision (A Maximizar):**
Porcentaje de patrones positivos predichos como positivos, frente al total de patrones predichos como positivos.

$$Precision = \frac{TP}{TP+FP}$$

Métricas para Clasificación Bi-clase

TP Rate, TN Rate, Precision, FP Rate, F-Measure

- **FP Rate (A Minimizar):**

Porcentaje de patrones negativos predichos como positivos.
Equivale a (1-Specificity).

$$FPRate = \frac{FP}{FP+TN}$$

- **FN Rate (A Minimizar):**

Porcentaje de patrones positivos predichos como negativos.

$$FNRate = \frac{FN}{FN+TP}$$

- **F-Measure o F-Score (A Maximizar):**

Combina las métricas *Recall* y *Precision*.

$$F - Measure = \frac{2*Precision*Recall}{Precision+Recall}$$

$$F - Measure = \frac{2TP}{2TP+FP+FN}$$

| | | Predicción | |
|------------|-------|--------------------|--------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

Métricas para Clasificación Bi-clase

Resumen gráfico en la matriz de confusión Bi-clase:

| | | Predicción | |
|------|----------------|----------------|----------------|
| | | C _P | C _N |
| Real | C _P | TP | FN |
| | C _N | FP | TN |

Accuracy

| | | Predicción | |
|------|----------------|----------------|----------------|
| | | C _P | C _N |
| Real | C _P | TP | FN |
| | C _N | FP | TN |

TP Rate (Recall)

| | | Predicción | |
|------|----------------|----------------|----------------|
| | | C _P | C _N |
| Real | C _P | TP | FN |
| | C _N | FP | TN |

FP Rate y
Specificity

| | | Predicción | |
|------|----------------|----------------|----------------|
| | | C _P | C _N |
| Real | C _P | TP | FN |
| | C _N | FP | TN |

Precision

| | | Predicción | |
|------|----------------|----------------|----------------|
| | | C _P | C _N |
| Real | C _P | TP | FN |
| | C _N | FP | TN |

F-measure

Métricas para Clasificación Multiclas

¿Cómo obtener los valores de las métricas anteriores para cada clase en clasificación Multiclas?

| | | Predicción | |
|------------|-------|---------------------------|---------------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

- Se pueden obtener todas las métricas anteriores pero siempre en función de una de las clases, que es la que se considera la positiva, contra el resto.

Una clase contra el resto en la matriz de confusión

| Clase predicha | | |
|----------------|----|----|
| a | b | c |
| 49 | 1 | 0 |
| 0 | 47 | 3 |
| | 2 | 48 |

| Clase predicha | | |
|----------------|----|----|
| a | b | c |
| 49 | 1 | 0 |
| 0 | 47 | 3 |
| | 2 | 48 |

| Clase predicha | | |
|----------------|----|----|
| a | b | c |
| 49 | 1 | 0 |
| 0 | 47 | 3 |
| | 2 | 48 |

clase b

| Clase predicha | | |
|----------------|----|----|
| a | b | c |
| 49 | 1 | 0 |
| 0 | 47 | 3 |
| | 2 | 48 |

clase c

TP
FN
FP
TN

Curvas ROC (*Receiver Operating Characteristics*)

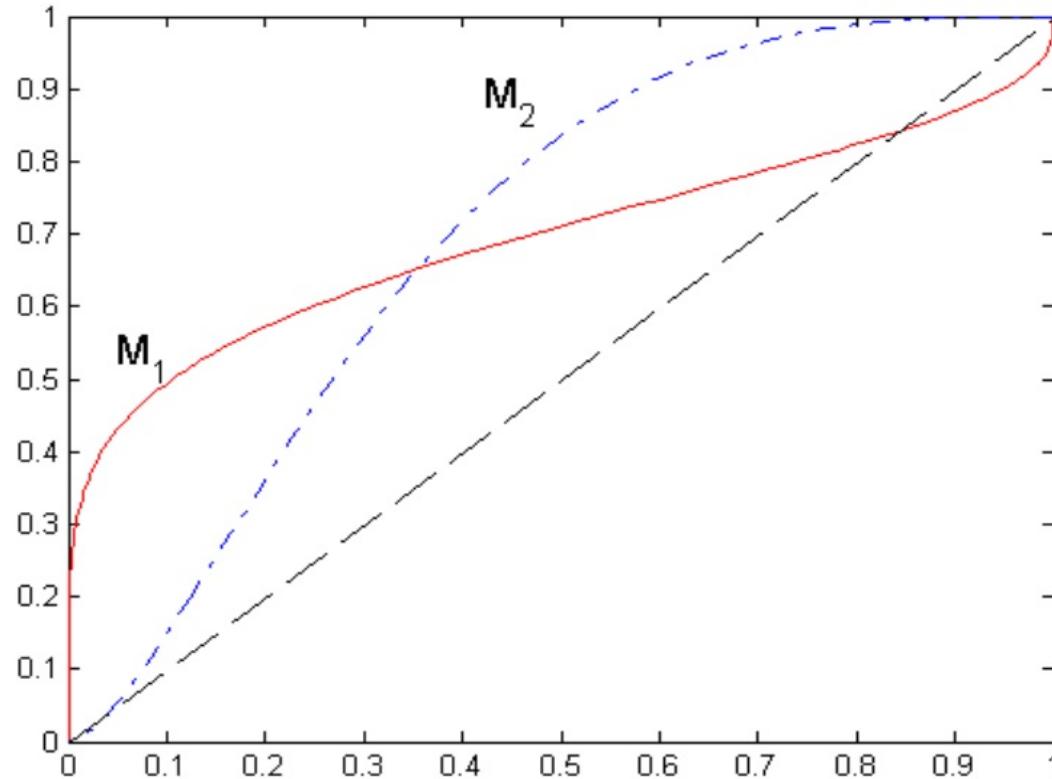
Otra medida para evaluar modelos **Bi-clase** es el área bajo una curva ROC.

Esta métrica se conoce como **AUC**, (*Area Under Curve*).

| | | Predicción | |
|------------|-------|--------------------|--------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

- Se usa para **modelos** que predigan la **probabilidad** de que un patrón pertenezca a la **clase Positiva**.
- **Umbral de discriminación:** Valor de probabilidad a partir del cual decidimos que un patrón es un positivo (clase positiva).
- Una ROC es la representación del ratio de verdaderos positivos (*TP*) frente al ratio de falsos positivos (*FP*), según se varía un umbral de discriminación.
- **Espacio bidimensional ROC:** *FPRate* eje X; *TPRate* eje Y.
- Cuanto más cerca se esté de la **diagonal** (**área cercana a 0.5**) de ambos ratios, **menos preciso** será el modelo o punto en el espacio ROC.

Curvas ROC (*Receiver Operating Characteristics*)



| | | Predicción | |
|------------|-------|--------------------|--------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

- En el ejemplo ningún modelo es consistentemente mejor que el otro: M_1 es mejor para *FP Rates* bajos, M_2 para *FP Rate* altos.
- Se podría calcular el AUC de cada uno.

Curvas ROC (*Receiver Operating Characteristics*)

| | | Predicción | |
|------------|-------|--------------------|--------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

- **TPRate:** Mide hasta qué punto el clasificador es **capaz de clasificar los casos positivos correctamente** de entre todos los casos positivos disponibles durante el *test*.
- **FPRate:** Mide **cuántos resultados clasificados como positivos son incorrectos** de entre todos los casos negativos disponibles durante el *test*.
- El mejor modelo posible de predicción se situaría en un punto en la esquina superior izquierda, o coordenada (0,1) del espacio ROC, representando un 100 % de sensibilidad (ningún FN) y un 100 % de especificidad (ningún FP). Un **modelo perfecto** tendrá un $AUC = 1$.

Curvas ROC (*Receiver Operating Characteristics*)

Ejemplo de cálculo de AUC en el espacio ROC:

- Suponga un modelo con 10 patrones para el conjunto de **testing**.
- Patrones de la clase positiva: 1, 2, 4, 8 y 10.
- Patrones de la clase negativa: 3, 5, 6, 7 y 9.
- Hay que buscar todos los posibles **umbrales de discriminación** que nos darían lugar a resultados diferentes.
- Se deben tener las **probabilidades de pertenencia** que da el modelo para que un patrón pertenezca a la clase positiva. En Weka se calcula automáticamente, no necesita aportar nada. En problemas multiclas, AUC se calcula considerando una clase frente a todas las demás.
- Se ordenan todos los patrones en **orden decreciente de la probabilidad de pertenencia a la clase positiva**.
- Se establece un **umbral de discriminación** por cada **probabilidad de pertenencia a la clase positiva**.

Curvas ROC (*Receiver Operating Characteristics*)

| | | Predicción | |
|------------|-------|---------------------------|---------------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

- Se calculan los valores de TP , FP , TN y FN , para cada umbral para el conjunto de patrones:
 - ▶ Si probabilidad \geq umbralActual: se clasifica en la **clase positiva** (TP o FP).
 - ▶ Si probabilidad $<$ umbralActual: se clasifica en la **clase negativa** (TN o FN).
- Columna **probabilidad**: Nos la da el modelo para cada patrón.
- Columna **clase**: La sabemos del conjunto de test para cada patrón.
- Se calculan los valores de $TPRate$ y $FPRate$:

$$TPRate = \frac{TP}{TP+FN}; FPRate = \frac{FP}{FP+TN}$$

| Patrón | Prob. | Clase | Clase | | + | + | - | + | - | - | - | + | - | + |
|--------|-------|-------|--------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 0.95 | + | Umbral | 1.00 | 0.95 | 0.93 | 0.87 | 0.85 | 0.85 | 0.85 | 0.76 | 0.53 | 0.43 | 0.25 |
| 2 | 0.93 | + | TP | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 5 |
| 3 | 0.87 | - | FP | 0 | 0 | 0 | 1 | 1 | 2 | 3 | 4 | 4 | 5 | 5 |
| 4 | 0.85 | + | TN | 5 | 5 | 5 | 4 | 4 | 3 | 2 | 1 | 1 | 0 | 0 |
| 5 | 0.85 | - | FN | 5 | 4 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 0 |
| 6 | 0.85 | - | TPRate | 0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 | 0.6 | 0.6 | 0.8 | 0.8 | 1 |
| 7 | 0.76 | - | FPRate | 0 | 0 | 0 | 0.2 | 0.2 | 0.4 | 0.6 | 0.8 | 0.8 | 1 | 1 |
| 8 | 0.53 | + | | | | | | | | | | | | |
| 9 | 0.43 | - | | | | | | | | | | | | |
| 10 | 0.25 | + | | | | | | | | | | | | |

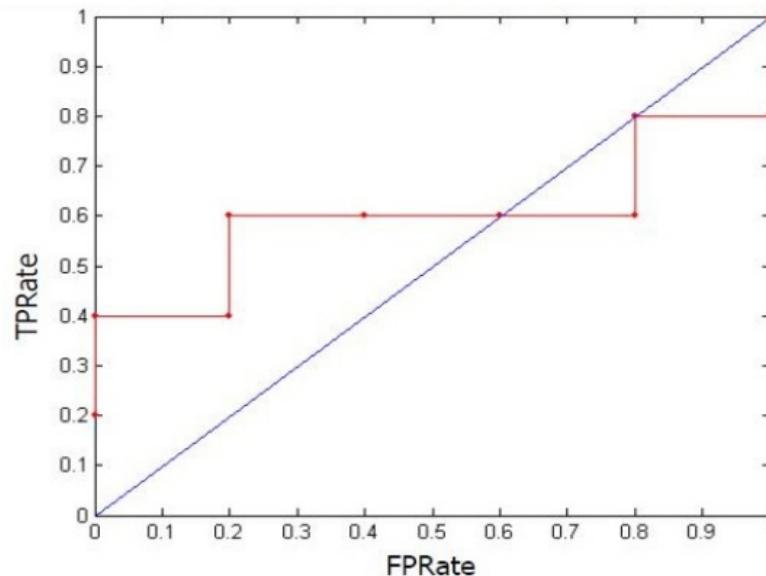
Curvas ROC (*Receiver Operating Characteristics*)

- Ya se puede dibujar la curva ROC y calcular su área:

| Patrón | Prob. | Clase |
|--------|-------|-------|
| 1 | 0.95 | + |
| 2 | 0.93 | + |
| 3 | 0.87 | - |
| 4 | 0.85 | + |
| 5 | 0.85 | - |
| 6 | 0.85 | - |
| 7 | 0.76 | - |
| 8 | 0.53 | + |
| 9 | 0.43 | - |
| 10 | 0.25 | + |

| Clase | + | + | - | + | - | - | + | - | + | |
|--------|------|------|------|------|------|------|------|------|------|------|
| Umbral | 1.00 | 0.95 | 0.93 | 0.87 | 0.85 | 0.85 | 0.76 | 0.53 | 0.43 | 0.25 |
| TP | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 5 |
| FP | 0 | 0 | 0 | 1 | 1 | 2 | 3 | 4 | 5 | 5 |
| TN | 5 | 5 | 5 | 4 | 4 | 3 | 2 | 1 | 0 | 0 |
| FN | 5 | 4 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 0 |
| TPRate | 0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 | 0.6 | 0.8 | 0.8 | 1 |
| FPRate | 0 | 0 | 0 | 0.2 | 0.2 | 0.4 | 0.6 | 0.8 | 1 | 1 |

| | | Predicción | |
|------------|----------------|--------------------|--------------------|
| | | C _P | C _N |
| Clase real | C _P | TP: True positive | FN: False negative |
| | C _N | FP: False positive | TN: True negative |



Curvas ROC (*Receiver Operating Characteristics*)

Ejemplo de cálculo para el *umbral=0,76*:

| | | Predicción | |
|------------|----------------|--------------------|--------------------|
| | | C _P | C _N |
| Clase real | C _P | TP: True positive | FN: False negative |
| | C _N | FP: False positive | TN: True negative |

- TP = 3 (patrones: 1, 2 y 4)
- FP = 4 (" : 3, 5, 6 y 7)
- TN = 1 (" : 9)
- FN = 2 (" : 8 y 10)

► Matriz de confusión

| | | Clase predicha | |
|------------|---|----------------|---|
| | | 3 | 2 |
| Clase real | 3 | 1 | 2 |
| | 4 | | 1 |

$$\blacktriangleright TPRate = \frac{TP}{TP+FN} = \frac{3}{3+2} = 0,6$$

$$\blacktriangleright FPRate = \frac{FP}{FP+TN} = \frac{4}{4+1} = 0,8$$

Estadístico KAPPA (binario y multiclasé)

Kappa

- Compara la **concordancia** observada en un conjunto de datos por un modelo, respecto a la que **podría ocurrir por mero azar**.
- Se calcula de igual manera para problemas **binarios y multiclasé**.
- Puede tomar valores en el rango $[-1, 1]$. A maximizar.
 - ▶ -1 = Discordancia total, peor que una clasificación al azar.
 - ▶ 1 = Concordancia perfecta, sin azar.
 - ▶ > 0 = Mayor concordancia que la que se esperaría por el puro azar.
 - ▶ 0 = No existe relación, la concordancia observada coincide con la que ocurriría por puro azar.

Estadístico KAPPA

$$Kappa = \frac{p_o - p_e}{1 - p_e}$$

$$p_o = CCR = \frac{1}{n} \sum_{j=1}^J n_{jj} \quad p_e = \frac{1}{n^2} \sum_{j=1}^J n_{j\bullet} n_{\bullet j}$$

| | | Predicción | |
|------------|-------|---------------------------|---------------------------|
| | | C_P | C_N |
| Clase real | C_P | TP: True positive | FN: False negative |
| | C_N | FP: False positive | TN: True negative |

- n_{jj} es un elemento de la matriz de confusión.
- J es el número de clases.
- n es el número de patrones en *testing*.
- $n_{j\bullet}$ es la suma de todos los elementos de la fila j
- $n_{\bullet j}$ es la suma de todos los elementos de la columna j
- **Como ejercicio calcule el valor Kappa de alguna matriz anterior.**

Regresión
oooooo

Regresión Lineal
oooooooooooo

Clasificación
oooooooooooooooooooo

k-NN
●○○

Regresión Logística
oooooooooooo

Entregables
oooo

Bibliografía
○

Regresión

Regresión Lineal

Clasificación

K-NN

Regresión Logística

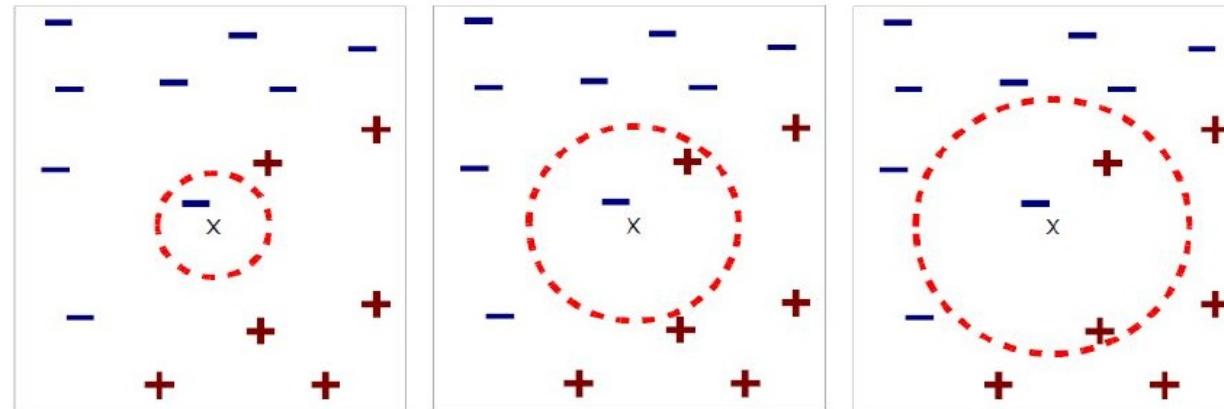
Entregables

Bibliografía

Clasificador: Vecinos más cercanos

- K-NN: Clasificador que se basa en los N patrones más cercanos a uno dado para etiquetar una clase.
- Un patrón “no visto” se coloca en el espacio multidimensional (atributos) de los patrones “vistos”, y se compara con los patrones “vistos” más cercanos (distancia euclídea).
 - ▶ https://es.wikipedia.org/wiki/K-vecinos_m%C3%A1s_cercanos
 - ▶ K demasiado pequeño (sensible a ruido).
 - ▶ K demasiado grande (el vecindario puede incluir puntos de otras clases).
- **KNN en Weka:** *weka/classifiers/Lazy*

K-NN es **IBk** (por defecto IB1, es decir, K=1)



(a) 1-nearest neighbor

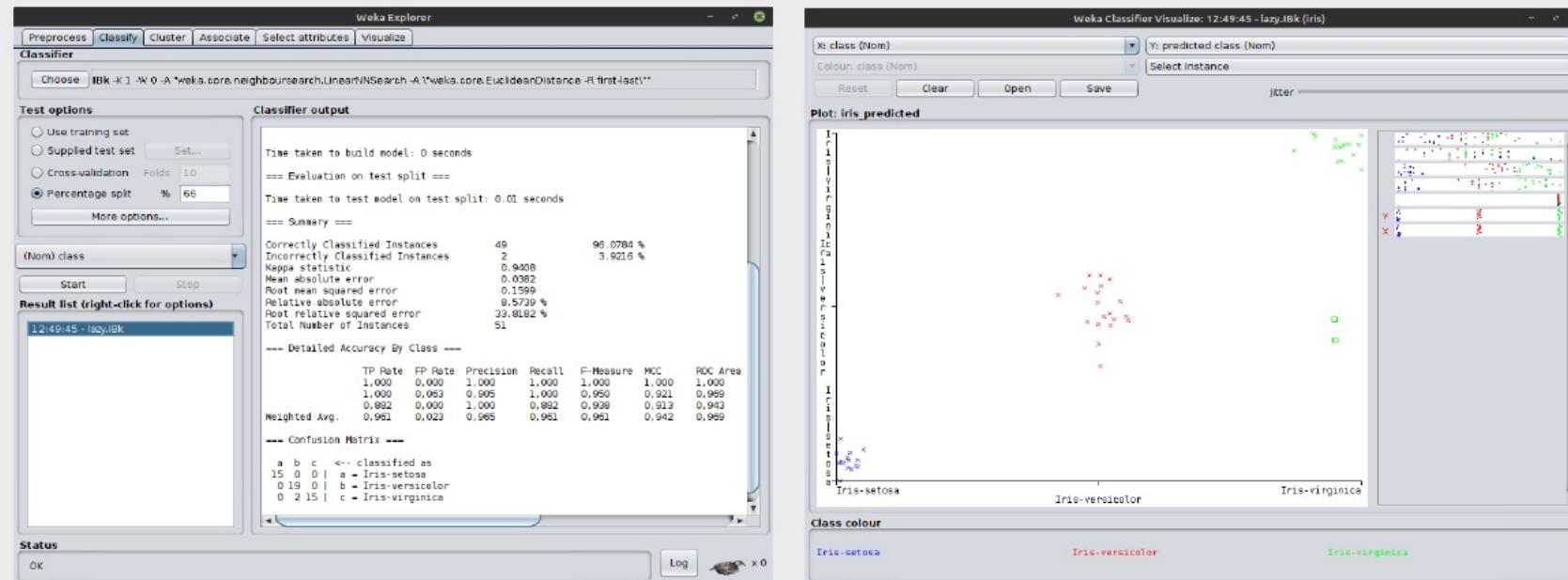
(b) 2-nearest neighbor

(c) 3-nearest neighbor

Visualización

Para visualizar errores de clasificación

- Botón derecho del ratón en “Result List” → Visualize classifier errors
- Comparar la grafica de los errores (los cuadrados) con la matriz de confusión, ¿se corresponden las filas y columnas?
No se corresponden exactamente, está traspuesta la matriz.



Regresión
oooooo

Regresión Lineal
oooooooooooo

Clasificación
oooooooooooooooooooo

k-NN
ooo

Regresión Logística
●oooooooooooo

Entregables
oooo

Bibliografía
o

Regresión

Regresión Lineal

Clasificación

K-NN

Regresión Logística

Entregables

Bibliografía

Regresión Logística

Regresión Logística

- Se usa en problemas donde la **variable dependiente es nominal**. Es para **CLASIFICACIÓN**.
- La **Regresión Lineal** forma parte de la ecuación de la **Regresión Logística**.
- https://es.wikipedia.org/wiki/Regresi%C3%B3n_log%C3%ADstica.
- Dos tipos, dependiendo de cómo se realice la optimización:
 - ▶ **Regresión Logística Simple:** Uso de heurística, algoritmo *Logitboost*.
 - ▶ **Regresión Logística:** Uso de máxima verosimilitud.

Regresión Logística Simple y Regresión Logística en Weka

- **Regresión Logística Simple:** *classifiers/functions/SimpleLogistic*
- **Regresión Logística:** *classifiers/functions/Logistic*

Salidas por clase en Simplelogistic y Logistic de Weka

Modelos de **regresión logistica** para un problema de **k=3** clases:

- SimpleLogistic

$$f_1(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$$

$$f_2(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$$

$$f_3(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$$

Función Softmax

$$p_1(\mathbf{x}, \hat{\theta}) = \frac{e^{f_1(\mathbf{x}, \hat{\theta})}}{\sum_{k=1}^K e^{f_k(\mathbf{x}, \hat{\theta})}}$$

$$p_2(\mathbf{x}, \hat{\theta}) = \frac{e^{f_2(\mathbf{x}, \hat{\theta})}}{\sum_{k=1}^K e^{f_k(\mathbf{x}, \hat{\theta})}}$$

$$p_3(\mathbf{x}, \hat{\theta}) = \frac{e^{f_3(\mathbf{x}, \hat{\theta})}}{\sum_{k=1}^K e^{f_k(\mathbf{x}, \hat{\theta})}}$$

- Logistic

$$f_1(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$$

$$f_2(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$$

Función Softmax

$$p_1(\mathbf{x}, \hat{\theta}) = \frac{e^{f_1(\mathbf{x}, \hat{\theta})}}{1 + \sum_{k=1}^{K-1} e^{f_k(\mathbf{x}, \hat{\theta})}}$$

$$p_2(\mathbf{x}, \hat{\theta}) = \frac{e^{f_2(\mathbf{x}, \hat{\theta})}}{1 + \sum_{k=1}^{K-1} e^{f_k(\mathbf{x}, \hat{\theta})}}$$

?

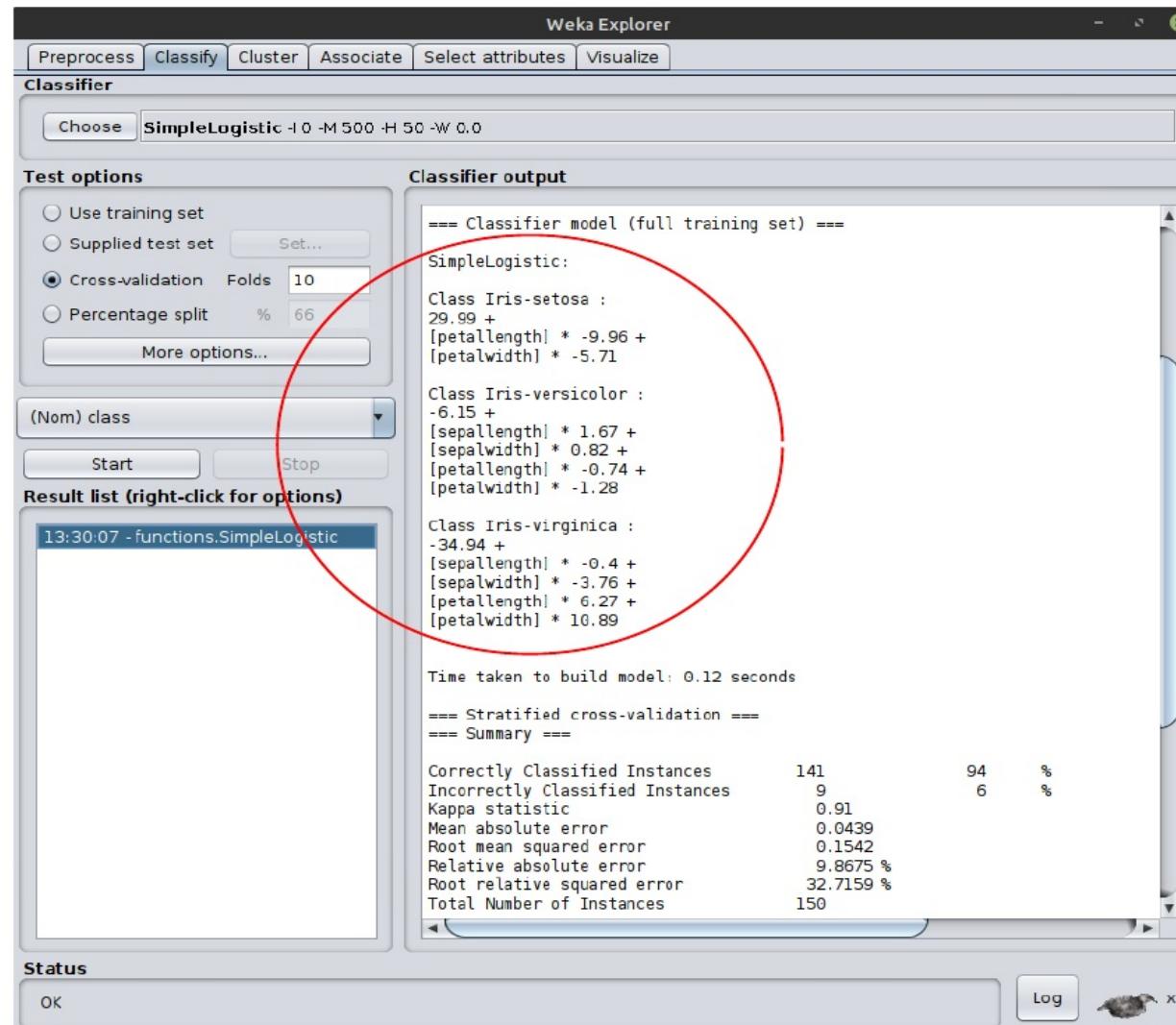
$$p_3(\mathbf{x}, \hat{\theta}) = 1 - \sum_{k=1}^{K-1} p_k(\mathbf{x}, \hat{\theta}) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{f_k(\mathbf{x}, \hat{\theta})}}$$

Salidas por clase en Simplelogistic y Logistic de Weka

- Para clasificación habrá **un modelo de regresión para cada una de las clases**. Se corresponde con las salidas $f_i(x, \hat{\theta})$ mostradas anteriormente,
- Como la **salida** de un modelo de regresión lineal es numérica, lo que haremos será **transformarla en un valor de probabilidad**.
- Para transformar en probabilidad se usa la función **Softmax** $\left(\frac{e^{(f_k)}}{\sum_{k=1}^K e^{(f_k)}} \right)$, que indicará la **probabilidad de pertenencia de un patrón i a una clase**.
https://es.wikipedia.org/wiki/Funci%C3%B3n_SoftMax
- **SimpleLogistic:**
 - ▶ Calcula la probabilidad con *Softmax* para cada clase.
- **Logistic:**
 - ▶ Calcula la probabilidad con *Softmax* para cada clase excepto la última, que será 1-resto.
 - ▶ En Weka por tanto no aparecerá la función de regresión para la última clase.

Modelo por clase para Iris con Simplelogistic de Weka

Regresión Logística Simple para Iris: classifiers/functions/SimpleLogistic



Métricas de salida para Iris con Simplelogistic en Weka

→ ¿Qué podría decir de las métricas de salida para Iris usando *Simplelogistic*?

The screenshot shows the Weka Explorer interface with the following details:

- Classifier:** SimpleLogistic -I 0 -M 500 -H 50 -W 0.0
- Test options:** Cross-validation Folds 10
- Result list:** 13:45:45 - functions.SimpleLogistic
- Classifier output:**
 - Time taken to build model: 0.18 seconds
 - ==== Stratified cross-validation ===
 - ==== Summary ===

| | Correctly Classified Instances | 94 | % |
|---------------------------|--------------------------------|----|---|
| Use training set | 141 | 94 | % |
| Supplied test set | 9 | 6 | % |
| Cross-validation Folds 10 | 141 | 94 | % |
| Percentage split % 66 | 9 | 6 | % |

 - Kappa statistic 0.91
 - Mean absolute error 0.0439
 - Root mean squared error 0.1542
 - Relative absolute error 9.8675 %
 - Root relative squared error 32.7159 %
 - Total Number of Instances 150
 - ==== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|-----------------|---------|---------|-----------|--------|-----------|-------|----------|----------|-----------------|
| Iris-setosa | 1,000 | 0,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | Iris-setosa |
| Iris-versicolor | 0,860 | 0,020 | 0,956 | 0,860 | 0,905 | 0,864 | 0,984 | 0,979 | Iris-versicolor |
| Iris-virginica | 0,960 | 0,070 | 0,873 | 0,960 | 0,914 | 0,871 | 0,988 | 0,970 | Iris-virginica |
| Weighted Avg. | 0,940 | 0,030 | 0,943 | 0,940 | 0,940 | 0,912 | 0,991 | 0,983 | |

 - ==== Confusion Matrix ====

| | a | b | c | classified as |
|---|----|----|----|---------------------|
| a | 50 | 0 | 0 | a = Iris-setosa |
| b | 0 | 43 | 7 | b = Iris-versicolor |
| c | 0 | 2 | 48 | c = Iris-virginica |
- Status:** OK

Métricas de salida para Iris con Simplelogistic en Weka

→ ¿Qué podría decir de las métricas de salida para Iris usando *Simplelogistic*?

- El valor de CCR es muy alto con un 94 %, es decir, acierta el 94 % de los patrones del conjunto de test. *A priori* buen clasificador para este problema.
- ¿Clasifica bien todas las clases? En este caso si, Iris-setosa tiene un TP-Rate de 1 (100 %), Iris-versicolor un TP-Rate de 0.860 (86 %) que es la que peor se clasifica y se ve además en la matriz de confusión, e Iris-virginica un TP-Rate de 0.960 (96 %). Los valores de F-measure también son cercanos a 1.
- Kappa es tambien cercano a 1, con lo cual es buen modelo.
- El resto de métricas se suelen usar más en regresión, para clasificación son menos interpretables ya que se van haciendo sumas de los errores en función de si se comete (1) o no se comete (0), es decir, de si se acierta la clase o no, pero igualmente son valores bajos, por tanto buen modelo de nuevo.
- La matriz de confusión muestra que hay 7 lirios versicolor que se clasifican como virginica y que hay 2 liros virginica que se clasifican como versicolor, que son los 9 patrones que hay mal clasificados sobre el conjunto de test.
- Al igual se se hacía con los β en regresión, se puede hacer una interpretación de qué características influyen más en la predicción de cada clase → Si **normaliza** previamente la base de datos tendrá mejores interpretaciones.

Modelo por clase para Iris con Logistic de Weka

Regresión Logística para Iris: classifiers/functions/Logistic

The screenshot shows the Weka Explorer interface with the following details:

- Classifier:** Choose Logistic -R 1.0E-8 -M 1 -num-decimal-places 4
- Test options:** Cross-validation Folds 10 (radio button selected)
- Result list (right-click for options):** 13:31:53 - functions.Logistic
- Classifier output:**
 - == Classifier model (full training set) ==
 - Logistic Regression with ridge parameter of 1.0E-8
 - Coefficients...

| Variable | Class | Iris-setosa | Iris-versicolor |
|-------------|-------|-------------|-----------------|
| sepallength | | 21.8065 | 2.4652 |
| sepalwidth | | 4.5648 | 6.6809 |
| petallength | | -26.3083 | -9.4293 |
| petalwidth | | -43.887 | -18.2859 |
| Intercept | | 8.1743 | 42.637 |

 - Odds Ratios...

| Variable | Class | Iris-setosa | Iris-versicolor |
|-------------|-------|-----------------|-----------------|
| sepallength | | 2954196659.8836 | 11.7653 |
| sepalwidth | | 96.0426 | 797.0304 |
| petallength | | 0 | 0.0001 |
| petalwidth | | 0 | 0 |

 - Time taken to build model: 0.04 seconds
 - Stratified cross-validation ---
 - Summary ---
 - Correctly Classified Instances 144 96 %
 - Incorrectly Classified Instances 6 4 %
 - Kappa statistic 0.94
 - Mean absolute error 0.0287
 - Root mean squared error 0.1424
 - Relative absolute error 6.456 %
 - Root relative squared error 30.2139 %
 - Total Number of Instances 150
- Status:** OK

Métricas de salida para Iris con Logistic en Weka

→ ¿Qué podría decir de las métricas de salida para Iris usando *Logistic*?

Weka Explorer

Classifier

Choose Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

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)

18:11:33 - functions.Logistic

Classifier output

Time taken to build model: 0.02 seconds

==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances 144 96 %
Incorrectly Classified Instances 6 4 %
Kappa statistic 0.94
Mean absolute error 0.0287
Root mean squared error 0.1424
Relative absolute error 6.456 %
Root relative squared error 30.2139 %
Total Number of Instances 150

==== Detailed Accuracy By Class ====

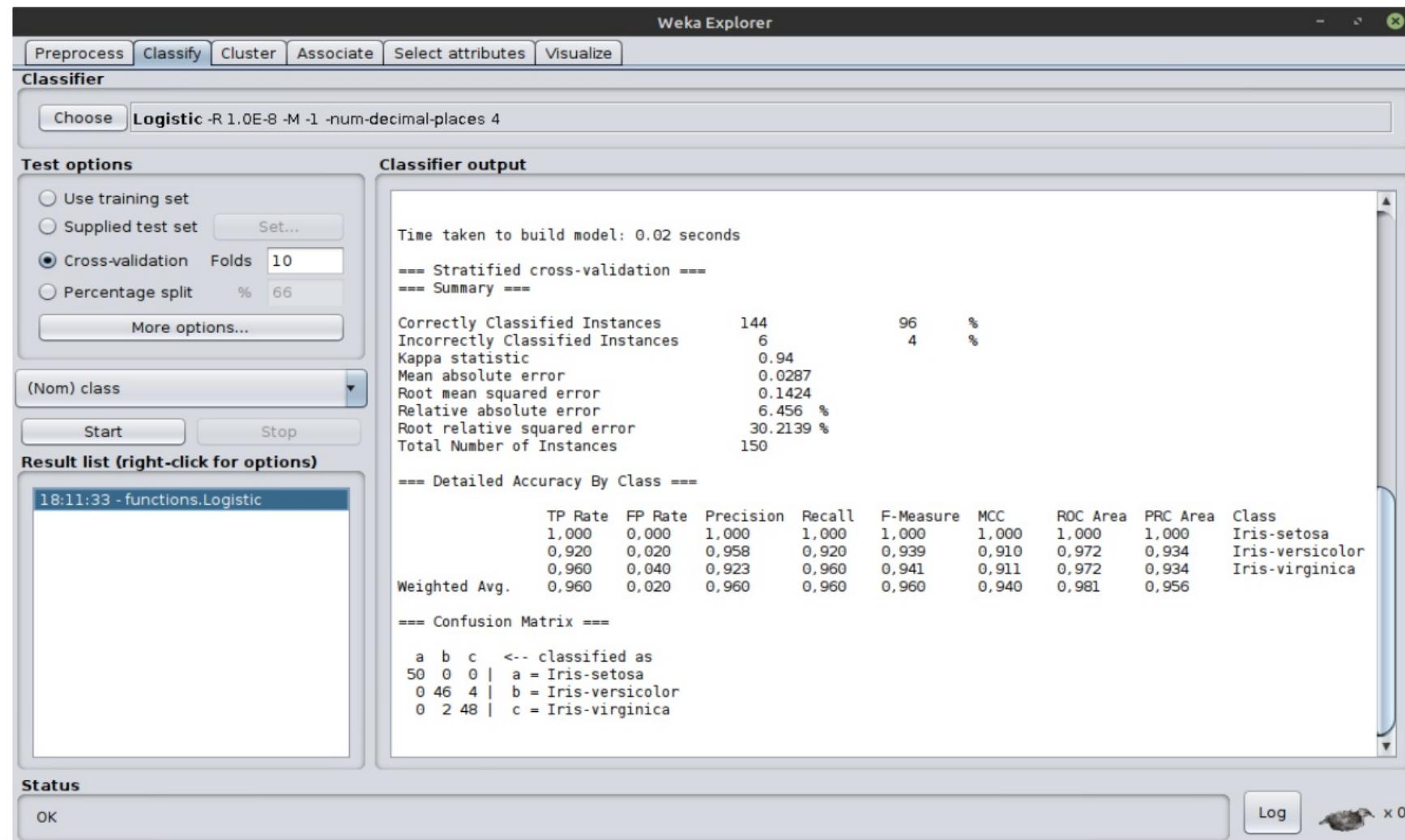
| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|-----------------|---------|---------|-----------|--------|-----------|-------|----------|----------|-----------------|
| Iris-setosa | 1,000 | 0,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | Iris-setosa |
| Iris-versicolor | 0,920 | 0,020 | 0,958 | 0,920 | 0,939 | 0,910 | 0,972 | 0,934 | Iris-versicolor |
| Iris-virginica | 0,960 | 0,040 | 0,923 | 0,960 | 0,941 | 0,911 | 0,972 | 0,934 | Iris-virginica |
| Weighted Avg. | 0,960 | 0,020 | 0,960 | 0,960 | 0,960 | 0,940 | 0,981 | 0,956 | |

==== Confusion Matrix ====
a b c <- classified as
50 0 0 | a = Iris-setosa
0 46 4 | b = Iris-versicolor
0 2 48 | c = Iris-virginica

Status

OK

Log x 0

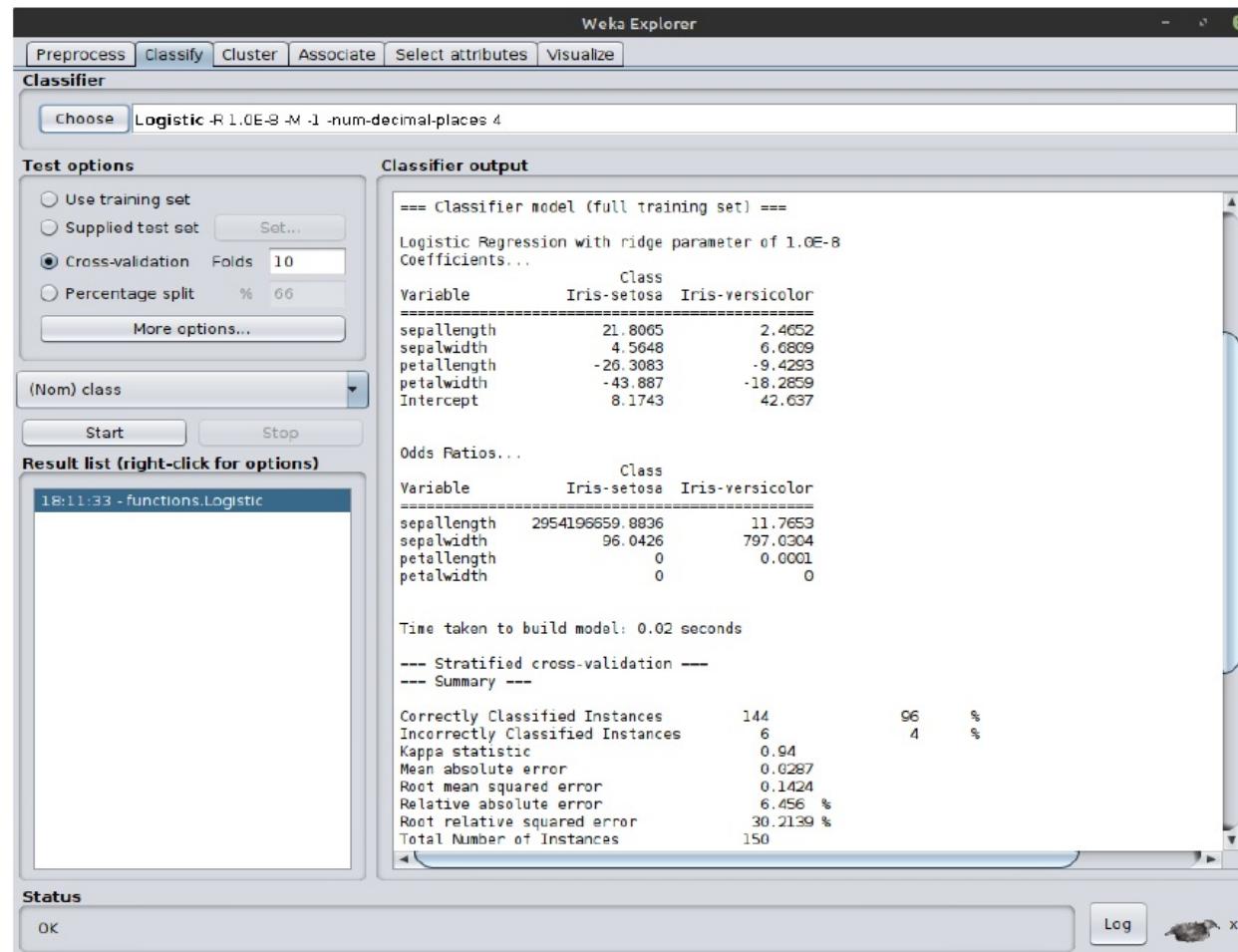


Métricas de salida para Iris con Logistic en Weka

- ¿Qué podría decir de las métricas de salida para Iris usando *Logistic*?
- El valor de CCR es muy alto con un 96 %, es decir, acierta el 96 % de los patrones del conjunto de test. *A priori* buen clasificador para este problema.
 - ¿Clasifica bien todas las clases? En este caso si, Iris-setosa tiene un TP-Rate de 1 (100 %), Iris-versicolor e Iris-virginica también tienen TP-Rate cercanos a 1. Los valores de F-measure también son buenos y cercanos a 1.
 - Kappa es tambien cercano a 1, con lo cual es buen modelo.
 - El resto de métricas se suelen usar más en regresión, para clasificación son menos interpretables ya que se van haciendo sumas de los errores en función de si se comete (1) o no se comete (0), es decir, de si se acierta la clase o no, pero igualmente son valores bajos, por tanto buen modelo de nuevo.
 - La matriz de confusión muestra que hay 4 lirios versicolor que se clasifican como virginica y que hay 2 liros virginica que se clasifican como versicolor, que son los 6 patrones que hay mal clasificados sobre el conjunto de test.
 - Al igual se hace con los β en regresión, se puede hacer una interpretación de qué características influyen más en la predicción de cada clase → Si **normaliza** previamente la base de datos tendrá mejores interpretaciones.

Métricas de salida para Iris con Logistic en Weka

- Además, en *Logistic* tenemos los valores de *odds ratio* (valores entre $0 - \infty$), que indican cuánto más probable es la pertenencia a una clase dado un cambio unitario en la variable X_i asociada con ese *odds ratio*.



Métricas de salida para Iris con Logistic en Weka

- Un **odds ratio** $X_i < 1$ significa que la probabilidad de pertenecer a la clase disminuirá por cada incremento unitario de la variable X_i , manteniéndose constantes las demás variables.

Ejemplo: Si **odds ratio** $X_i = 0,5$, la probabilidad de pertenecer a la clase se reducirá a la mitad con respecto a no pertenecer.

- Un **odds ratio** $X_i = 1$ significa que la probabilidad de pertenecer a la clase se mantendrá sin variación por cada incremento unitario de la variable X_i , manteniéndose constante las demás variables. No hay relación entre las variables, da igual que esté presente o no.
- Un **odds ratio** $X_i > 1$ significa que la probabilidad de pertenecer a la clase aumentará por cada incremento unitario de la variable X_i , manteniéndose constantes las demás variables.

Ejemplo: Si **odds ratio** $X_i = 2$, la probabilidad de pertenecer a la clase se duplicará con respecto a no pertenecer.

Ejemplo: Para apuestas sería ¡Las apuestas para que gane el Barsa están 2 a 1!

- Un **odds ratio** cercano a 0 o muy grande (∞) significa que una gran dependencia entre la variable X_i respecto a la probabilidad de pertenecer a la clase.
- Para Iris, las variables *petallength* y *petalwidth* son determinantes para discernir entre las clases, ya que tienen un valor de 0.

Regresión
oooooo

Regresión Lineal
oooooooooooo

Clasificación
oooooooooooooooooooo

k-NN
ooo

Regresión Logística
oooooooooooo

Entregables
●ooo

Bibliografía
o

Regresión

Regresión Lineal

Clasificación

K-NN

Regresión Logística

Entregables

Bibliografía

Entregables para el guión final

1. Escoja una de las bases de datos de clasificación para el trabajo de las dispuestas en Moodle (Breast cancer, Dermatology, Fantasmas, Glass, Vehicle, Wine, Zoo).

Se entiende que además de pasarla a formato .arff ya ha aplicado el preprocesamiento necesario en función del fichero “***Pistas sobre los datasets con posible preprocesamiento a simple vista.pdf***”, en el caso de que sea una de las bases de datos que lo requiera.

- ▶ Aplique preprocesamiento adicional (si se puede aplicar) sobre: 1) reemplazamiento de datos perdidos, 2) normalización y 3) paso de nominal a binario u ordinal a numérico.

Explique el preprocesamiento que haya llevado a cabo en los aspectos citados, y de no tener que hacerlo explique también por qué.

Entregables para el guión final

2. Con la base de datos escogida anteriormente, use el algoritmo de clasificación **KNN** (IBK en Weka) con un *10-fold crossvalidation*. Use un valor de vecinos $k=3$ dejando por defecto el resto de parámetros.

- ▶ Interprete la salida en cuanto a los valores de las métricas que proporciona Weka.

Tenga en cuenta si se clasifican bien todas las clases de su problema (*TP Rate* por clase) y fíjese también en la matriz de confusión.

Para explicar los resultados haga uso de tablas donde se muestren los valores que está interpretando.

Entregables para el guión final

3. Con la base de datos escogida anteriormente, ejecute el algoritmo *Simple-Logistic* con *10-fold crossvalidation*.
 - ▶ Analice los modelos obtenidos, las variables que podrían ser más influyentes (valores β), variables que no se usan y métricas. Use tablas para explicar los resultados de manera que haya una lectura legible.

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



Weka: The workbench for machine learning, 2019.
[https://www.cs.waikato.ac.nz/ml/weka/.](https://www.cs.waikato.ac.nz/ml/weka/)

Regresión
oooooo

Regresión Lineal
oooooooooooo

Clasificación
oooooooooooooooooooo

k-NN
ooo

Regresión Logística
oooooooooooo

Entregables
oooo

Bibliografía
●

¿Preguntas?