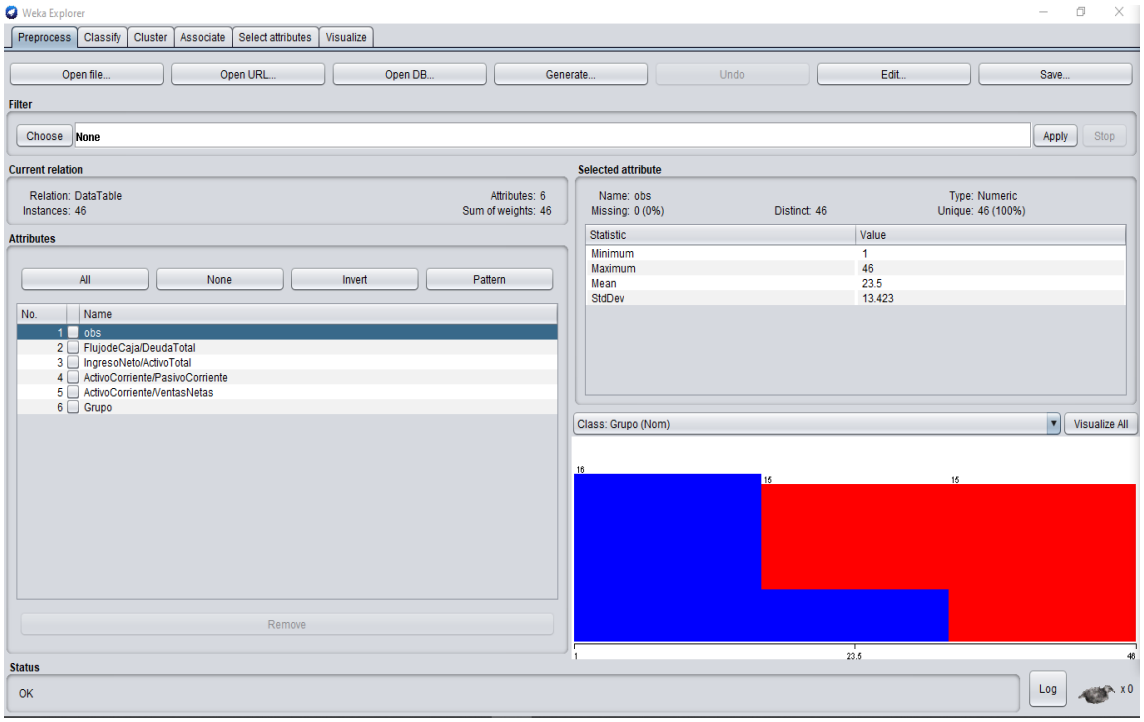
19 Mayo 2020. Tarea Support Vector Machines. Weka. Parte3

Archivo: empresas2.arff

1. **Abrir el archivo**
2. **Eliminar el atributo obs. ¿Porqué se elimina obs para la clasificación?**
3. **Classify, classifiers, functions, SMO (SMO es un algoritmo de optimización usado para entrenar un SVM)**
4. **Confirmar que la variable objetivo es Grupo**
5. **Clic para editar los argumentos**
6. **En filterType seleccionar 'No normalization/standarization'**
7. **Clic en Start**
8. **Presentar la matriz de confusión. Cuántos elementos quedan mal clasificados.**
9. **Ejecutar nuevamente modificando en kernel exponent=2**
10. **Presentar la matriz de confusión. Cuántos elementos quedan mal clasificados.**

1.

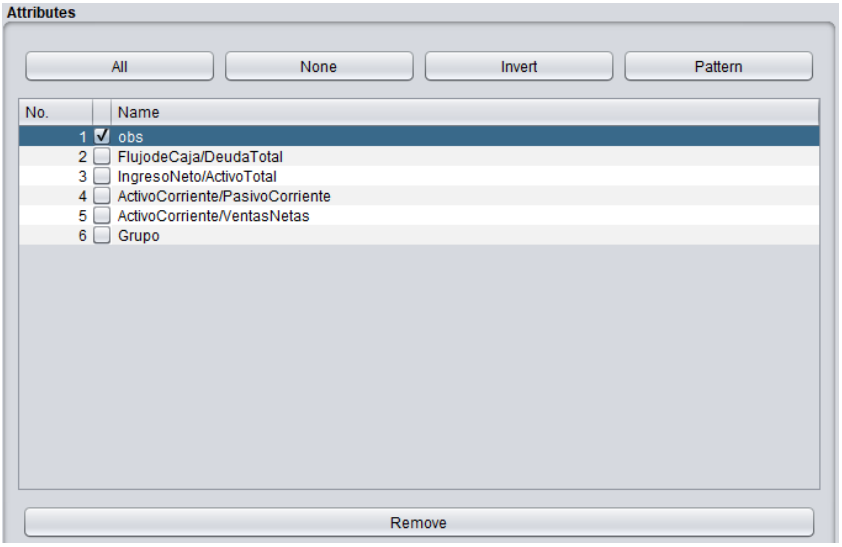
En primer lugar ejecutamos la herramienta Weka. En la plantalla principal del programa elegimos la opción “Explorer”, y en la nueva ventana que nos aparecerá seleccionamos “Open file…” y buscamos el archivo “empresas.txt” y lo abrimos. Deberíamos llegar a esta pantalla una vez realizados los pasos anteriores:



2.

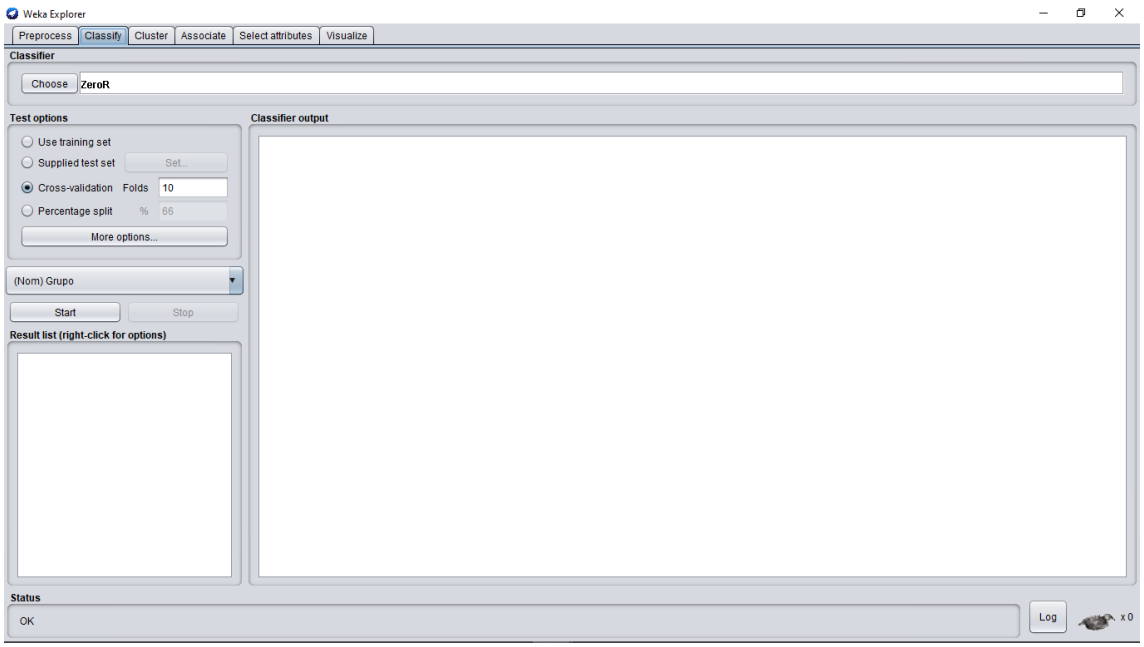
Ahora, en la lista de atributos, seleccionaremos “obs” y la eliminaremos de los atributos, ya que el atributo “obs” ya que las empresas vienen ordenadas por las que quebraron y las que no quebraron. Por lo tanto están perfectamente clasificados, y esta atributo realmente nunca lo recibiríamos así.

Para eliminarlo lo seleccionamos en la lista, y en la parte inferior de la susodicha observaremos un botón que reza “Remove”. Al clickarlo quitaremos el atributo de la lista.

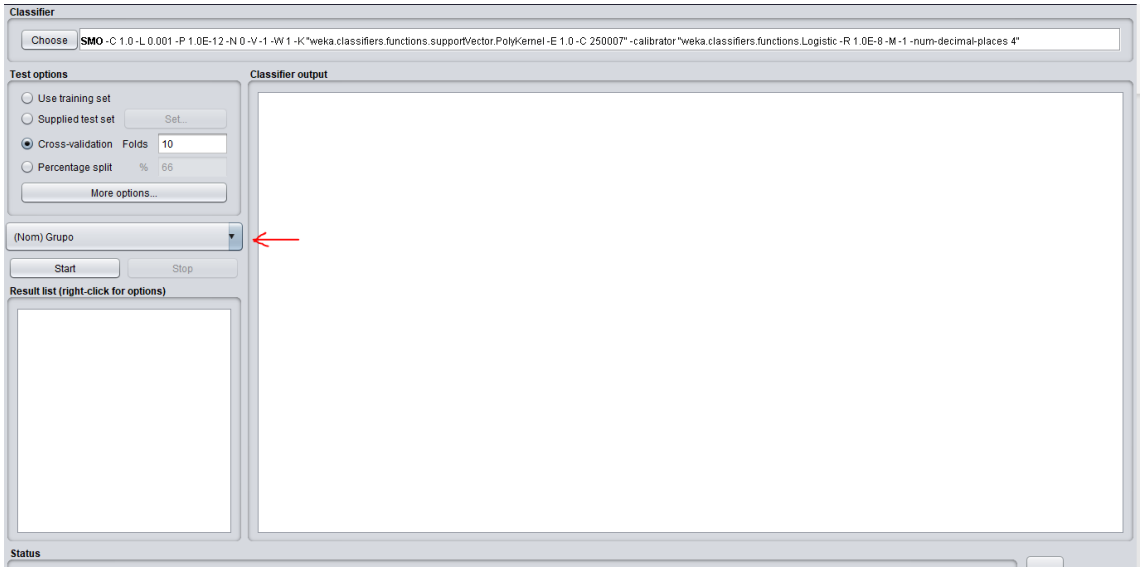


3.

Una vez eliminado el atributo “obs”, nos iremos a la pestaña “Clasify”, que se encuentra en la parte superior de la ventana de “Weka Explorer”. Hacemos click y llegamos a esta pantalla:



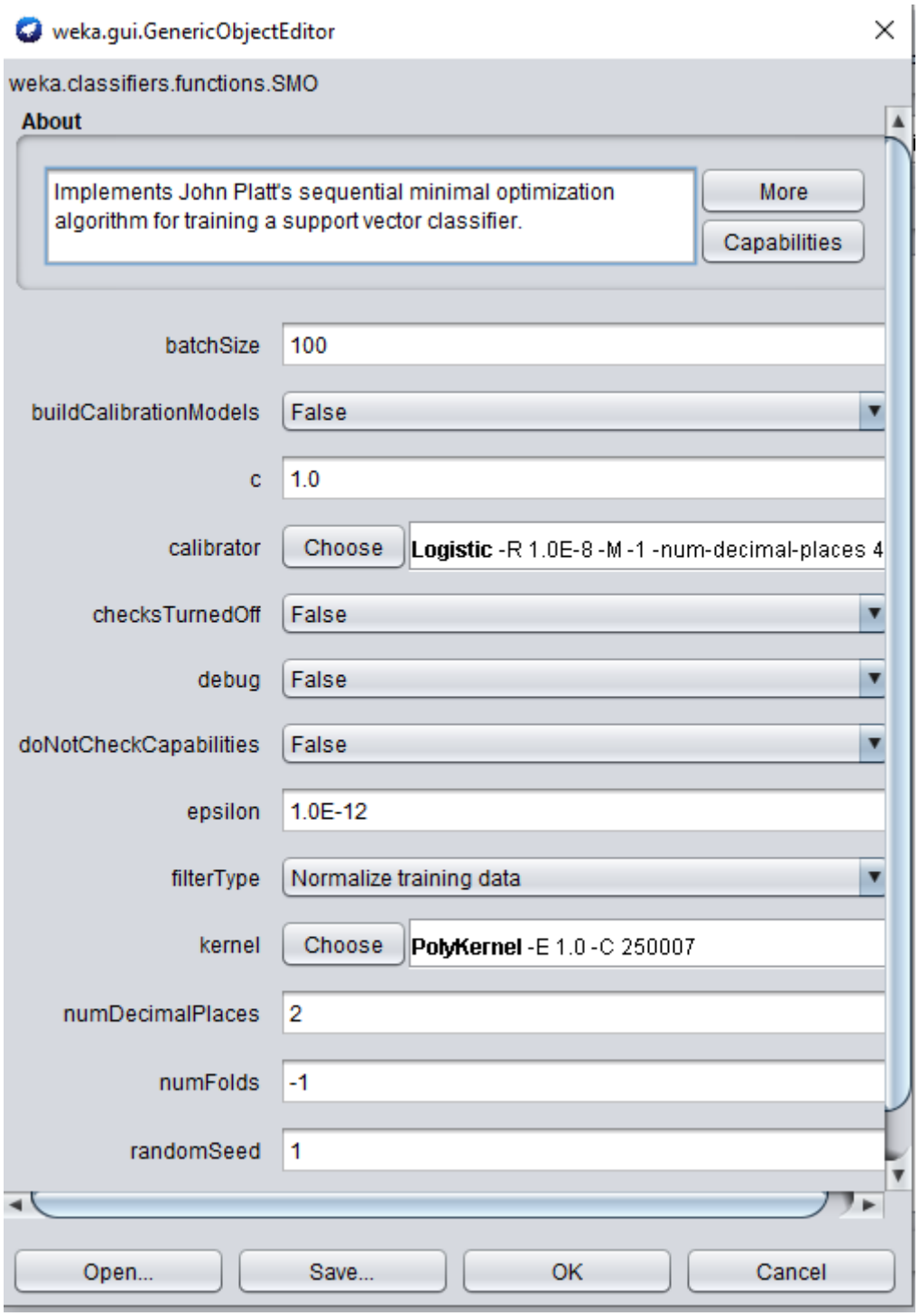
Vamos a hacer click en “Choose” y dentro de la carpeta “classifiers”, en la subcarpeta “functions” seleccionamos el algoritmo “SMO”, ya que aquí se encuentra el algoritmo “dSVM”



Nos aseguramos de que el atributo que queda explicado es “Grupo”.

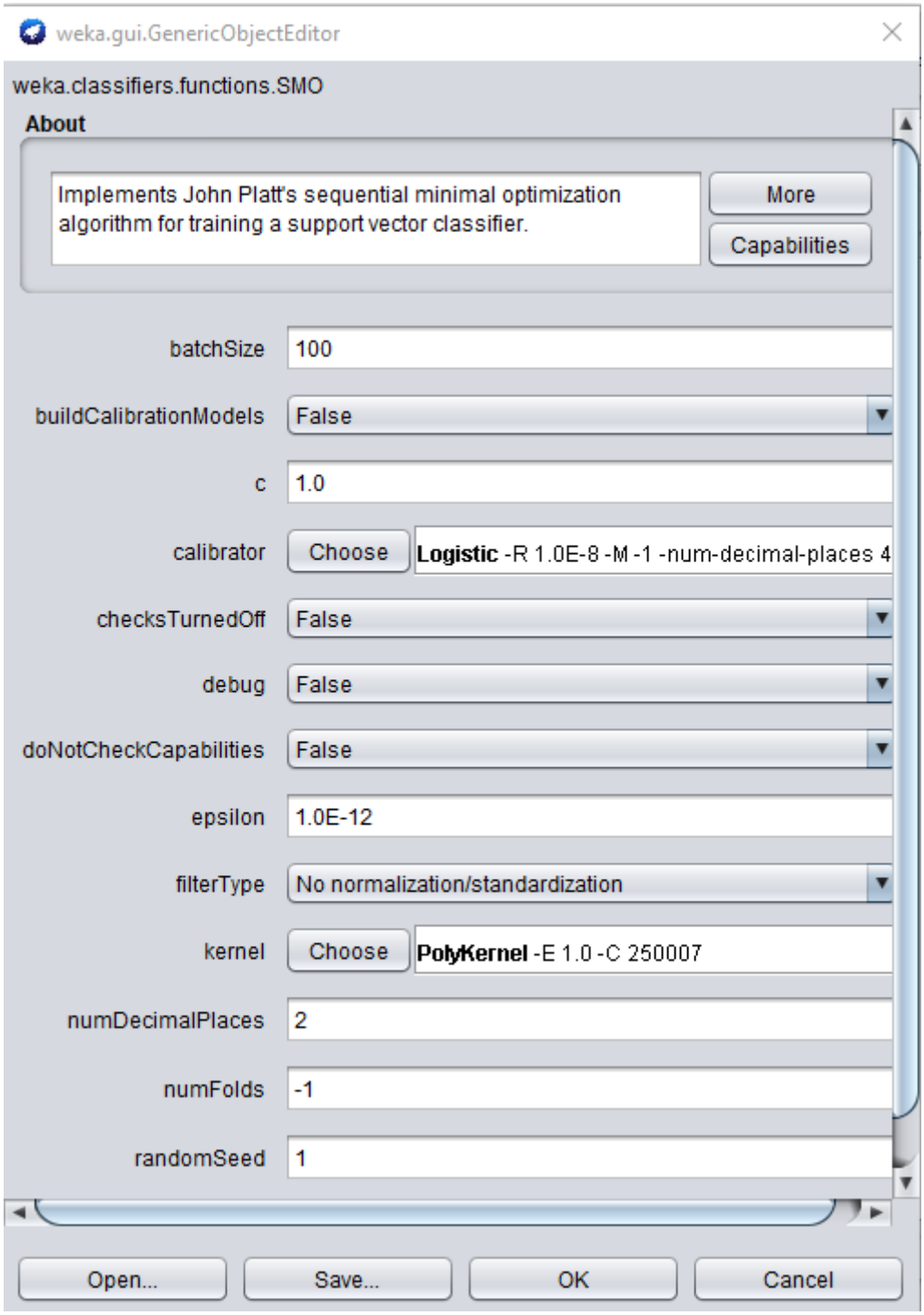
5, 6 y 7.

Hacemos click en el algoritmo “SMO -C1.0 -L.0.001-…” para editar sus argumentos:



En ocasiones los datos se normalizan con el objetivo de llevarlo a una unidad de medida común. Pero nosotros no lo vamos a normalizar, ya que en las cajas negras es usual que no sea necesario normalizar los datos ya que no tenemos por qué entender.

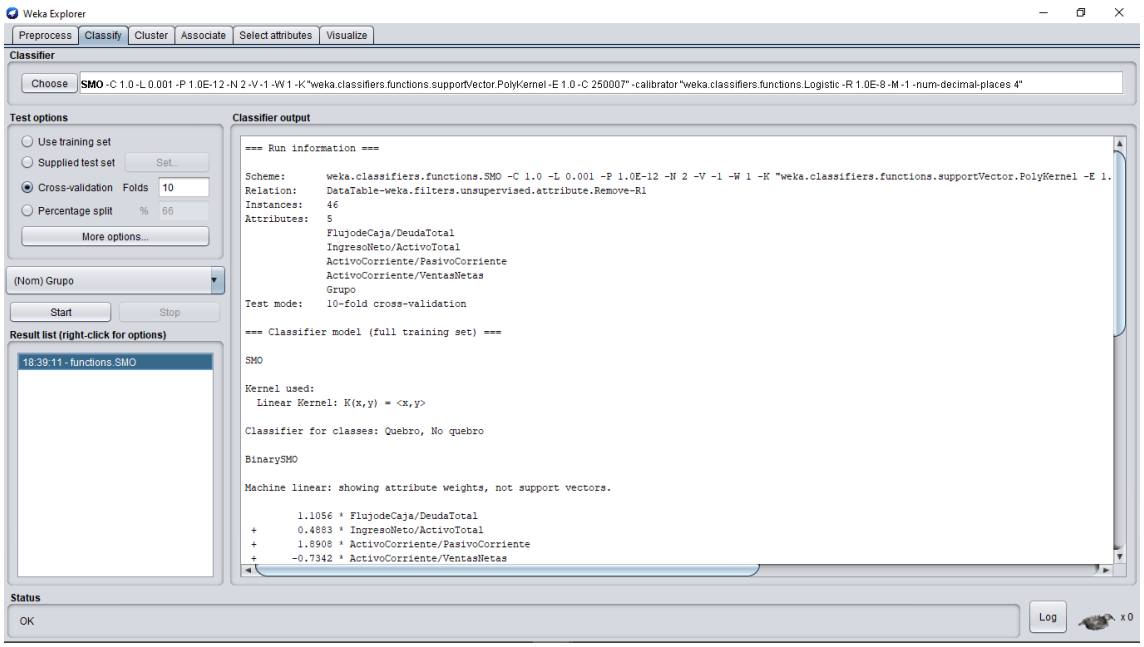
Por tanto, en la opción que reza “filterType” la vamos a cambiar por “No normalization/standardization”.



Una vez hecho esto estamos en condiciones de ejecutar el algoritmo, por lo que pulsamos el botón “Start”.

8.

Una vez hemos ejecutado el algoritmo, se nos presenta una ventana con los datos recogidos en el proceso de este.



Si exploramos entre los datos, llegaremos a la matriz de confusión, en la que podemos observar cuantos elementos están bien clasificados, es decir, tanto en la realidad como en el modelo creado mediante el algoritmo “SVM”, 18 empresas quebraron frente a las 3 empresas que no quebraron en la realidad pero el modelo predijo que si quebrarían.

```
Time taken to build model: 0.05 seconds

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

Correctly Classified Instances          40               86.9565 %
Incorrectly Classified Instances         6               13.0435 %
Kappa statistic                          0.7371
Mean absolute error                      0.1304
Root mean squared error                  0.3612
Relative absolute error                 26.2195 %
Root relative squared error             72.3445 %
Total Number of Instances               46

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,857    0,120    0,857      0,857    0,857      0,737    0,869     0,800     Quebro
               0,880    0,143    0,880      0,880    0,880      0,737    0,869     0,840     No quebro
Weighted Avg.  0,870    0,132    0,870      0,870    0,870      0,737    0,869     0,821

=== Confusion Matrix ===

  a  b   <-- classified as
 18  3 |  a = Quebro
  3 22 |  b = No quebro
```

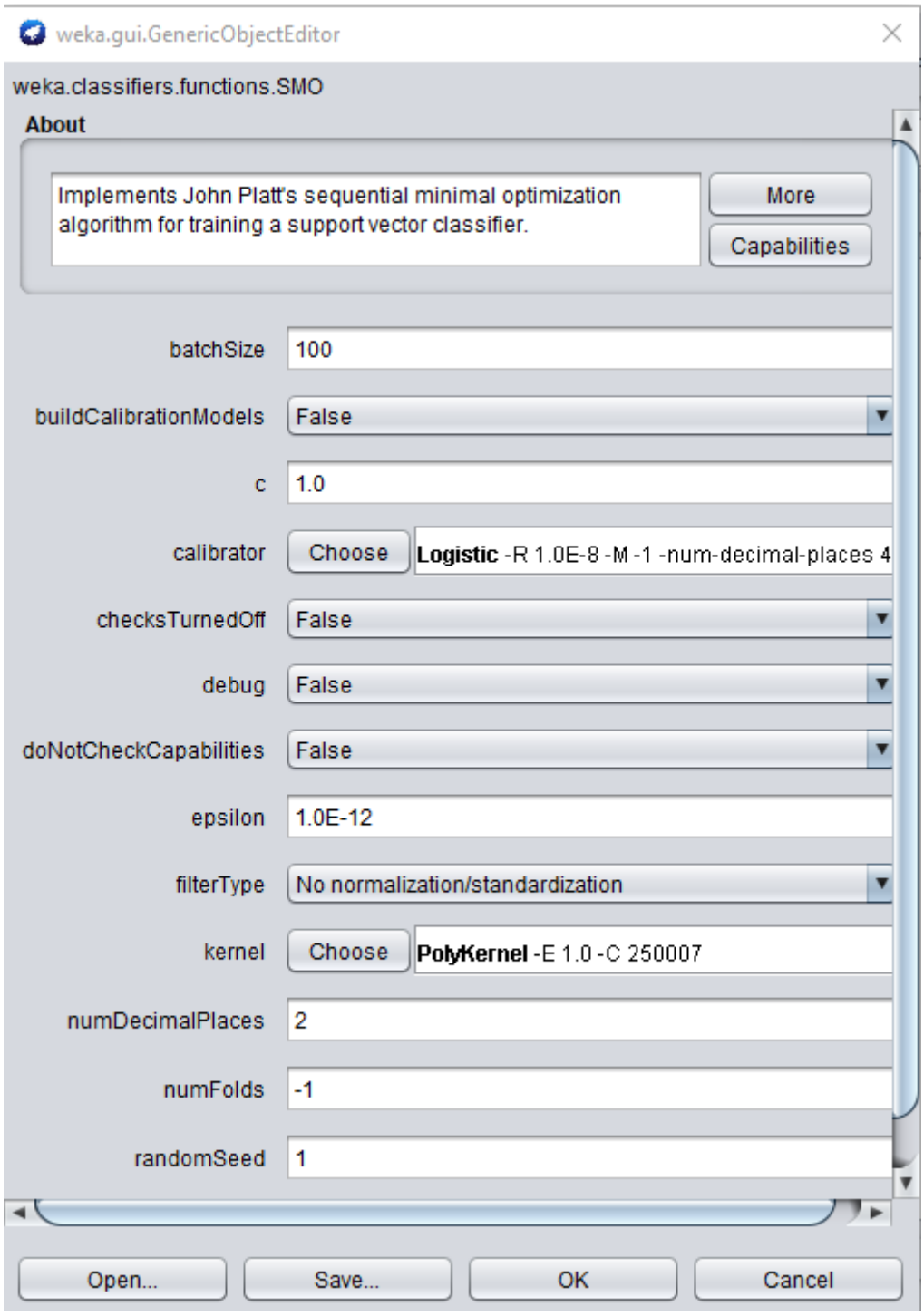
En el caso de las empresas que no quebraron, el modelo y la realidad coincidieron en 22 empresas, frente a las 3 empresas que quedaron mal clasificadas, pues el algoritmo predijo que quebrarían, pero en la realidad no fue así.

Por tanto, tenemos 40 empresas bien clasificado frente a 6 empresas que quedaron mal clasificadas.

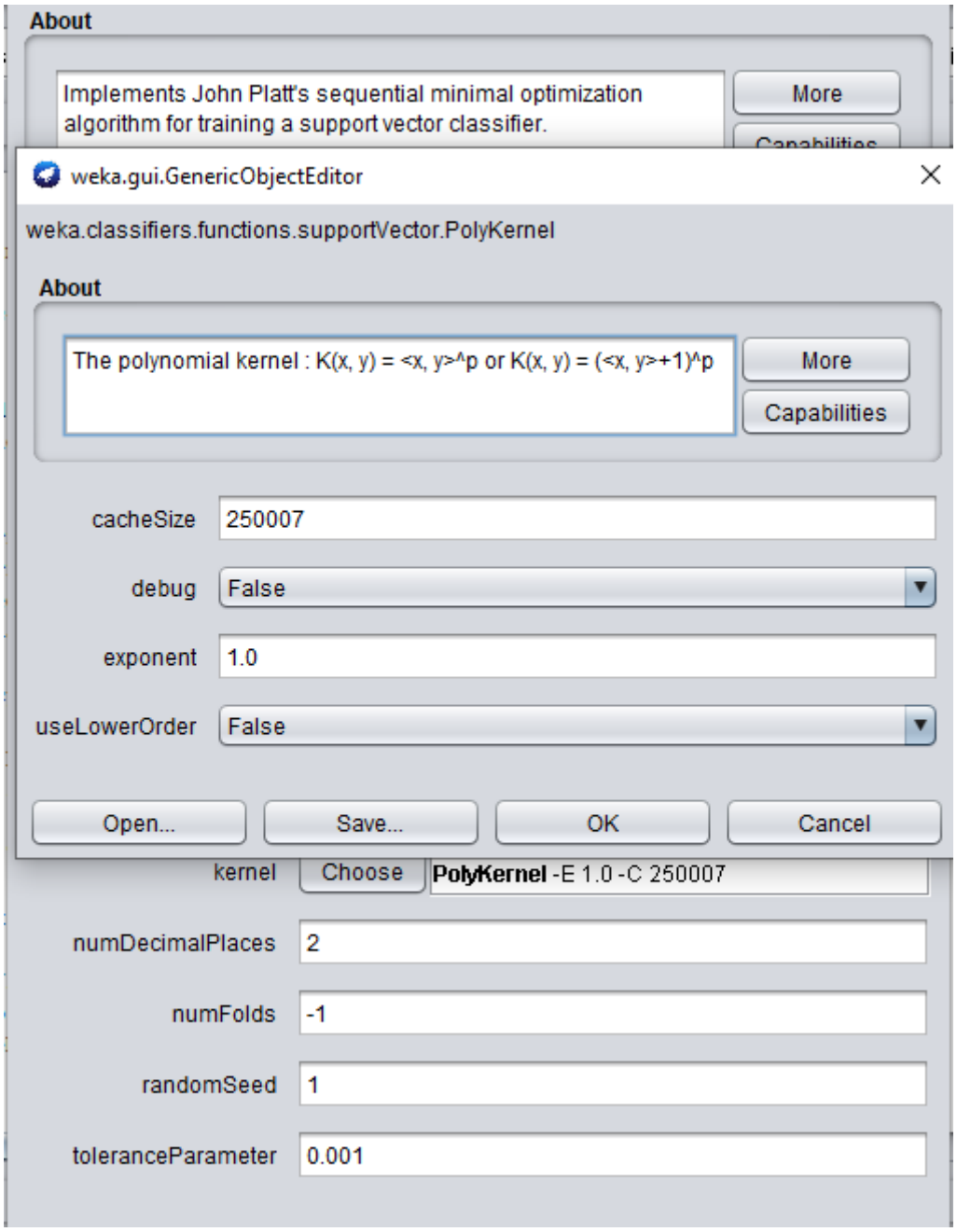
9.

Ahora vamos a modificar uno de los parámetros de algoritmo “SMO”, concretamente el “Kernel exponent”. Este exponente es el que tendrán los atributos de entrada en la ecuación.

Para ello hacemos click en “SMO -C1.0 -L.0.001-…” para modificar los parámetros del algoritmo.



Dónde dice “Kernel Chose” haremos click llegando a la siguiente ventana:

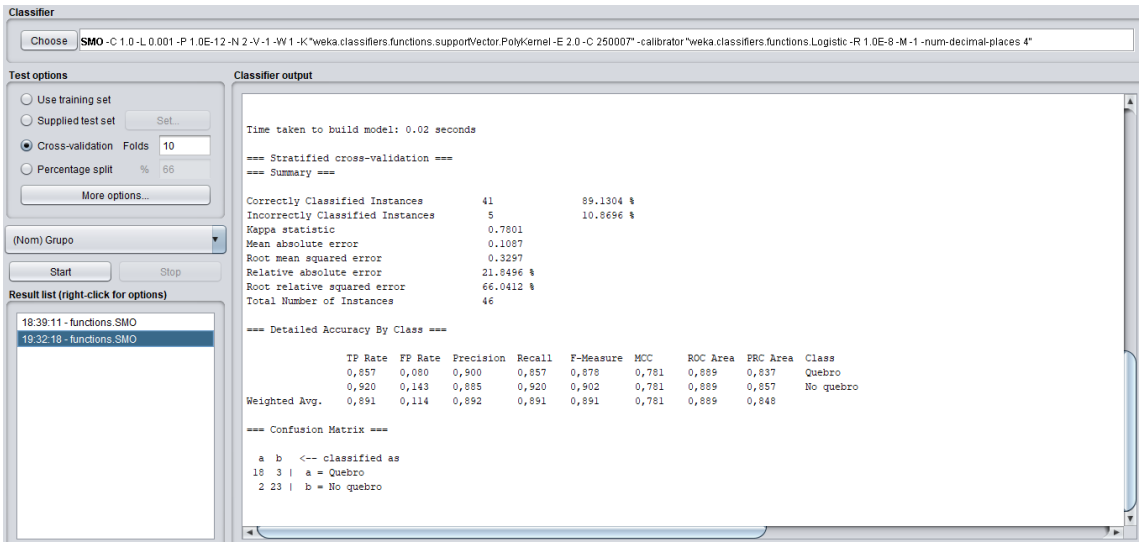


En el lugar donde pone “exponent”, cambiaremos el “1.0” por “2.0” para que el exponente de los términos de la función generada por el algoritmo sea 2.

Tras hacer esto pulsamos “Ok” y ejecutamos el algoritmo pulsando “Start”.

10.

Tras ejecutar el algoritmo llegaremos a la siguiente pantalla:



Aquí se nos presentan los datos recabados por el algoritmo. A continuación, presentamos los coeficientes de los atributos de entrada y el término independiente:

```
SMO

Kernel used:
  Poly Kernel: K(x,y) = <x,y>^2.0

Classifier for classes: Quebro, No quebro

BinarySMO

      1       * <0.4785 0.091 1.2444 0.1847 > * X]
 -        1       * <0.1454 0.05 1.8762 0.2723 > * X]
 -        1       * <0.0724 -0.0076 1.3723 0.3361 > * X]
 +        1       * <0.2907 0.0597 1.8381 0.3786 > * X]
 -        1       * <0.0109 0.0011 2.1495 0.6969 > * X]
 -        1       * <0.0713 0.0205 1.3124 0.2497 > * X]
 +        1       * <-0.0173 0.0233 2.0538 0.3484 > * X]
 -        0.3248 * <0.0451 0.0263 1.6756 0.9494 > * X]
 +        0.8167 * <0.2029 0.0792 1.9936 0.3018 > * X]
 -        0.7614 * <0.1227 0.1055 1.1434 0.1655 > * X]
 -        1       * <0.3703 0.1098 1.9941 0.3828 > * X]
 +        1       * <0.1703 0.0695 1.7973 0.5174 > * X]
 +        1       * <0.0769 0.0195 2.0069 0.5304 > * X]
 -        0.7305 * <-0.0721 -0.093 1.4544 0.2589 > * X]
 +        1       * <0.1398 -0.0312 0.4611 0.2643 > * X]
 -        2.0462

Number of support vectors: 15

Number of kernel evaluations: 772 (86.468% cached)
```

Podemos observar que tenemos 15 vectores de soporte.

Por otro lado, en esta misma pantalla, pero un poco más abajo, vemos la matriz de confusión.

```
=== Confusion Matrix ===

  a  b   <-- classified as
 18  3 |  a = Quebro
  2 23 |  b = No quebro
```

En esta matriz podemos ver que tenemos un total de 5 empresas mal clasificados y 41 empresas bien clasificadas.

Por tanto, hemos mejorado la precisión del algoritmo de predicción en este caso, aumentando el exponente de la función.