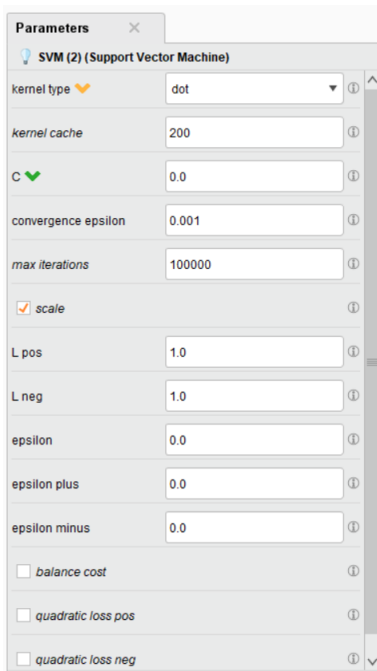


# UT4- PD4

Valores parámetro del operador SVM

Parámetros por defecto:



Parameter	Value
kernel type	dot
kernel cache	200
C	0.0
convergence epsilon	0.001
max iterations	100000
scale	<input checked="" type="checkbox"/>
L pos	1.0
L neg	1.0
epsilon	0.0
epsilon plus	0.0
epsilon minus	0.0
balance cost	<input type="checkbox"/>
quadratic loss pos	<input type="checkbox"/>
quadratic loss neg	<input type="checkbox"/>

- Kernel Type: Selecciona la función a ser usada como kernel en el modelo.
  - o Dot: El kernel punto es  $k(x,y) = x \cdot y$ , es el producto interno de  $x$  y  $y$ .
  - o Radial: Se define como  $\exp(-g \cdot ||x-y||^2)$  donde  $g$  es el gamma especificado por parámetro. Este gamma es crucial en la performance del algoritmo y debe ser ajustado a mano.
  - o Polynomial: Está definido por  $k(x,y) = (x \cdot y + 1)^d$  donde  $d$  es el grado del polinomio y se especifica por parámetro. Encajan en los problemas donde los datos de entrada están normalizados.
  - o Neural: Está definido por una red neuronal de dos capas  $\tanh(a \cdot x \cdot y + b)$  donde  $\alpha$  y  $b$  es la constante de interceptación. Estos parámetros pueden ser ajustados usando los parámetros  $a$  y  $b$ . Un valor común para  $\alpha$  es  $1/N$  con  $N$ = dimensión de los datos.
  - o Anova: Está definido por elevar a la potencia  $d$  la suma  $\exp(-g(x-y))$  donde  $g$  es gamma y  $d$  es grado. Gamma y  $d$  son parámetros ajustables.
  - o Epachnenikov: Es esta función  $(3/4)(1-u^2)$  para  $u$  entre  $-1$  y  $1$  y cero para  $u$  fuera del rango. Tiene dos parámetros ajustables kernel sigma1 y grado del kernel.
  - o Gaussian\_combination: Kernel de combinación gaussiana, tiene los parámetros ajustables: kernel sigma1, kernel sigma2 y kernel sigma3.
  - o Multiquadric: El kernel multiquadric está definido por la raíz cuadrada de  $||x-y||^2 + c^2$ . Tiene el parámetro ajustable kernel sigma1 y kernel sigma shift.
- Kernel\_cache: Especifica el tamaño del cache para las evaluaciones del kernel en megabytes.

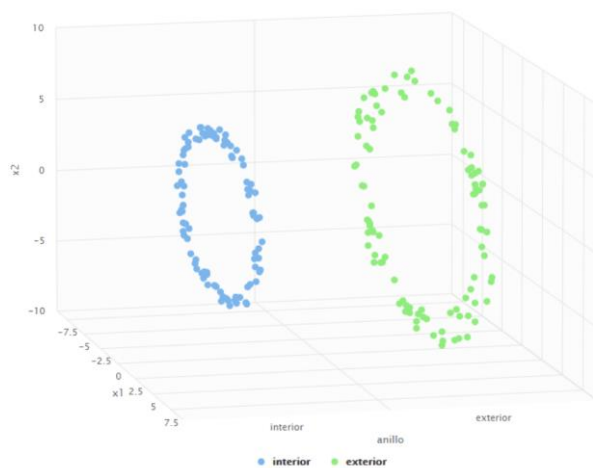
- C: Es la constante de complejidad que setea la tolerancia para la clasificación errónea. Un c más alto permite límites más permisivos y los valores bajos crean límites más fuertes. Una constante muy grande lleva al sobre ajuste del modelo y valores muy pequeños pueden resultar en sobre generalización.
- Convergence\_epsilon: Especifica la precisión en las condiciones KKT
- Max\_iterations: Es un parámetro optimizador. Especifica un número límite de iteraciones.
- Scale: Es un parámetro global. Si está seleccionado los valores ejemplo son escalados y los parámetros de escalación son guardados para un conjunto de testeo.
- L\_pos: Un factor para el factor de complejidad del SVM para ejemplos positivos. Este parámetro forma parte de la función de pérdida.
- L\_neg: Un factor para el factor de complejidad del SVM para ejemplos negativos. Este parámetro forma parte de la función de pérdida.
- Épsilon: Este parámetro especifica la constante de intensidad. No hay pérdida si la predicción recae a esto del valor real. Es parte de la función pérdida.
- Épsilon plus: Es un parámetro para la función pérdida. Especifica la epsilon para las desviaciones positivas solamente.
- Épsilon minus: Es un parámetro para la función pérdida. Especifica la epsilon para las desviaciones negativas solamente.
- Balance cost: Si seleccionado, adapta Cpos y Cneg a los tamaños relativos de las clases.
- Quadratic loss pos: Usa la pérdida cuadrática para desviaciones positivas. Este parámetro es parte de la función pérdida.
- Quadratic loss neg: Usa la pérdida cuadrática para desviaciones negativas. Este parámetro es parte de la función pérdida.

## Performance con los parámetros por defecto

accuracy: 46.67%

	true interior	true exterior	class precision
pred. interior	16	18	47.06%
pred. exterior	14	12	46.15%
class recall	53.33%	40.00%	

Es prácticamente como tirar a adivinar, el modelo no detecta relaciones y es predecible porque el kernel punto funciona con relaciones lineales.



Las instancias están separadas en dos clases bien delimitadas, pero en una forma circular. No es una relación lineal simple.

Performance con el kernel polinomial grado 2.

accuracy: 100.00%

	true interior	true exterior	class precision
pred. interior	30	0	100.00%
pred. exterior	0	30	100.00%
class recall	100.00%	100.00%	

En este caso el modelo puede detectar las regiones perfectamente porque el kernel polinomial es capaz de detectar líneas curvas en la separación del conjunto de datos.