

INSTITUTO SUPERIOR TÉCNICO

APRENDIZAGEM AUTOMÁTICA

---

## Laboratório 5 - Support Vector Machines

---

*Autores:*

Diogo Moura - nº 86976

Diogo Alves - nº 86980

*Turno:*

Terça 17h-18h30m

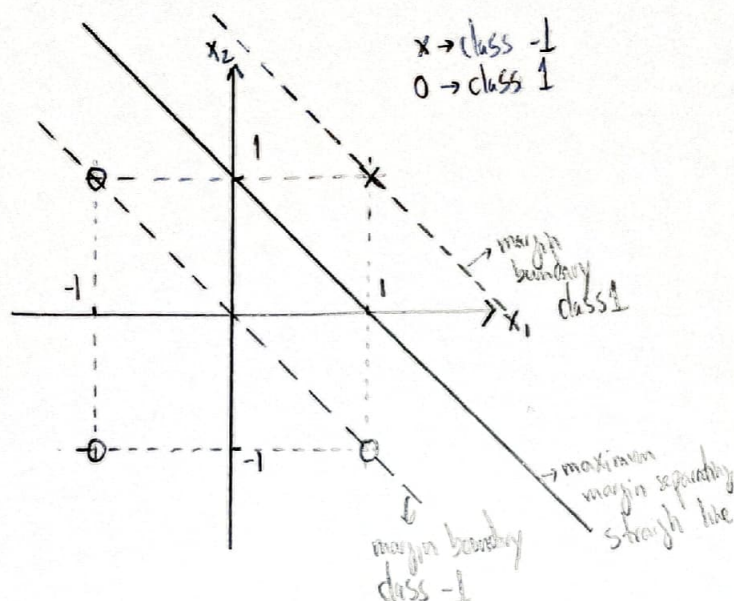
26 de Novembro de 2019



**TÉCNICO**  
LISBOA

## 2 - Two Simple examples

2.1. AND:



Support vectors:  $x^1 = (1, 1) \rightarrow \delta^1 = 1$   
 $x^2 = (1, -1) \rightarrow \delta^2 = -1$   
 $x^3 = (-1, 1) \rightarrow \delta^3 = -1$

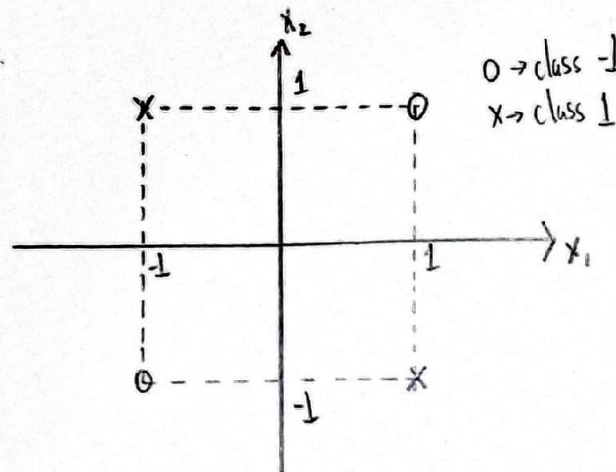
Tem que se verificar a equação:  
 $(w \cdot x^s + b) \delta^s = 1$   
 para todos os SV

$$\begin{cases} (w_1, w_2) \cdot (1, 1) + b \cdot 1 = 1 \\ (w_1, w_2) \cdot (1, -1) + b \cdot (-1) = 1 \\ (w_1, w_2) \cdot (-1, 1) + b \cdot (-1) = 1 \end{cases} \Leftrightarrow \begin{cases} w_1 + w_2 + b = 1 \\ -w_1 + w_2 - b = 1 \\ w_1 - w_2 - b = 1 \end{cases}$$

$$\Leftrightarrow \begin{cases} w_1 = 1 - b - w_2 \\ -1 + b + w_2 + w_2 - b = 1 \\ 1 - b - w_2 - w_2 - b = 1 \end{cases} \Leftrightarrow \begin{cases} w_1 = 1 \\ w_2 = 1 \\ b = -1 \end{cases}$$

2.2.

XOR:



As classes não são linearmente separáveis - não existe uma reta que consiga separar os pontos de cada classe

$$\varphi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}$$

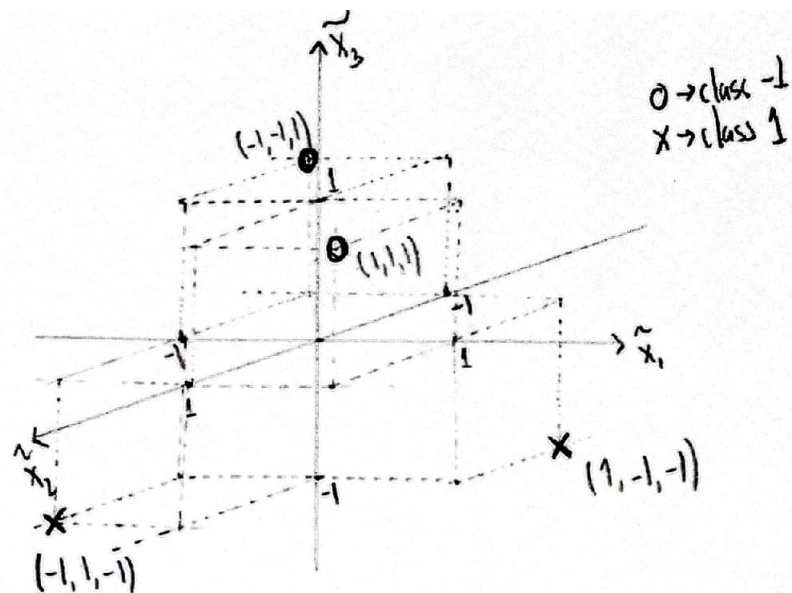
$$\text{Kernel: } K(x, y) = \varphi(x) \cdot \varphi(y) = \begin{bmatrix} x_1 & x_2 & x_1 x_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_1 y_2 \end{bmatrix} = x_1 y_1 + x_2 y_2 + x_1 x_2 y_1 y_2$$

$$= x \cdot y + x_1 x_2 y_1 y_2$$

2.3.

$x$		$\tilde{x}$			$d_{\text{XOR}}$
$x_1$	$x_2$	$x_1$	$x_2$	$x_1 x_2$	
-1	-1	-1	-1	1	-1
-1	1	-1	1	-1	1
1	-1	1	-1	-1	1
1	1	1	1	1	-1





Por inspeção, o plano de separação de classes será o plano  $\tilde{x}_3 = 0$ , do qual todos os pontos se encontram à mesma distância, pelo que todos são Support Vectors

Support Vectors:

$$\begin{aligned} \tilde{x}^1 &= (-1, -1, 1) \rightarrow d^1 = -1 \\ \tilde{x}^2 &= (-1, 1, -1) \rightarrow d^2 = 1 \\ \tilde{x}^3 &= (1, -1, -1) \rightarrow d^3 = 1 \\ \tilde{x}^4 &= (1, 1, 1) \rightarrow d^4 = -1 \end{aligned}$$

$$\begin{cases} (\tilde{w}_1 \tilde{x}^1 + b) \cdot d^1 = 1 \\ (\tilde{w}_1 \tilde{x}^2 + b) \cdot d^2 = 1 \\ (\tilde{w}_1 \tilde{x}^3 + b) \cdot d^3 = 1 \\ (\tilde{w}_1 \tilde{x}^4 + b) \cdot d^4 = 1 \end{cases}$$

$$\Leftrightarrow \begin{cases} \tilde{w}_1 + \tilde{w}_2 - \tilde{w}_3 - b = 1 \\ -\tilde{w}_1 + \tilde{w}_2 - \tilde{w}_3 + b = 1 \\ \tilde{w}_1 - \tilde{w}_2 - \tilde{w}_3 + b = 1 \\ -\tilde{w}_1 - \tilde{w}_2 - \tilde{w}_3 + b = 1 \end{cases}$$

$$\Leftrightarrow \begin{cases} \tilde{w}_1 = 1 + \tilde{w}_3 + b - \tilde{w}_2 \\ \tilde{w}_2 - 1 - \tilde{w}_3 - b + \tilde{w}_2 - \tilde{w}_3 + b = 1 \\ 1 + \tilde{w}_3 + b - \tilde{w}_2 - \tilde{w}_2 - \tilde{w}_3 + b = 1 \\ \tilde{w}_2 - 1 - \tilde{w}_3 - b - \tilde{w}_2 - \tilde{w}_3 - b = 1 \end{cases}$$

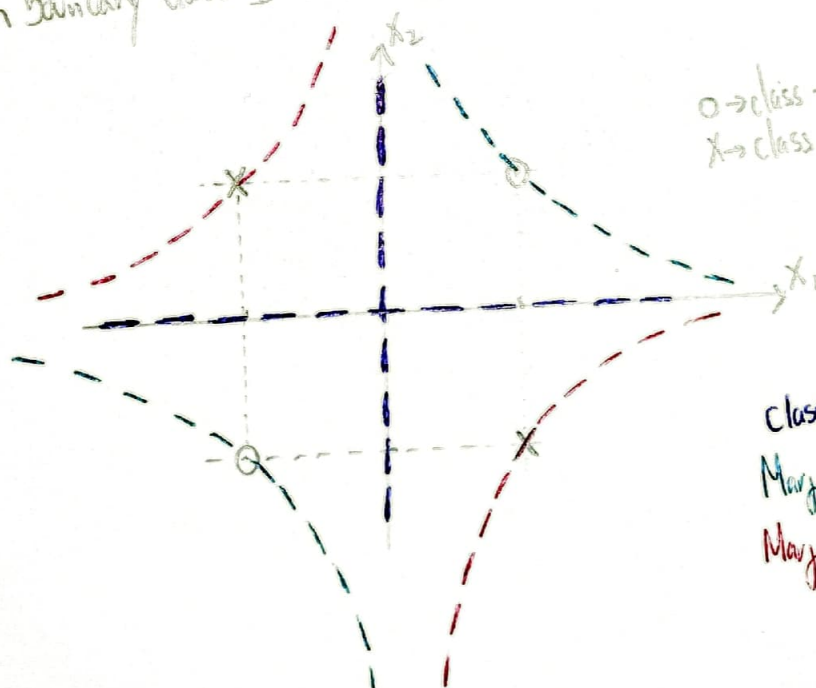
$$\Leftrightarrow \begin{cases} \tilde{w}_1 = 1 + \tilde{w}_3 + b - \tilde{w}_2 \\ \tilde{w}_2 = 1 + \tilde{w}_3 \\ \tilde{w}_2 = b \\ -\tilde{w}_2 - \tilde{w}_2 - \tilde{w}_3 - \tilde{w}_3 = 0 \end{cases} \Leftrightarrow \begin{cases} \tilde{w}_1 = 0 \\ \tilde{w}_3 = -1 \\ b = 0 \\ \tilde{w}_2 = 0 \end{cases}$$

2.4.

Classification Border:  $\tilde{w} \cdot \tilde{x} + b = 0 \Leftrightarrow$   
 $\Leftrightarrow \tilde{x}_1 \tilde{w}_1 + \tilde{x}_2 \tilde{w}_2 + \tilde{x}_3 \tilde{w}_3 + b = 0 \quad (=) \quad -\tilde{x}_3 = 0 \Leftrightarrow$   
 $\Leftrightarrow -x_1 x_2 = 0 \Leftrightarrow x_1 = 0 \vee x_2 = 0$

Margin boundary class -1:  $\tilde{w} \cdot \tilde{x} + b = 1 \Leftrightarrow -\tilde{x}_3 = -1 \Leftrightarrow x_2 x_1 = 1$   
 $\Leftrightarrow x_2 = \frac{1}{x_1}$

Margin boundary class 1:  $\tilde{w} \cdot \tilde{x} + b = -1 \Leftrightarrow -\tilde{x}_3 = 1 \quad (=) \quad x_2 = -\frac{1}{x_1}$



$\circ \rightarrow$  class -1  
 $\times \rightarrow$  class 1

classification Boundary  
 Margin boundary class -1  
 Margin boundary class 1

2.5.

Observando o gráfico, o output será 1 se se verificar a condição:

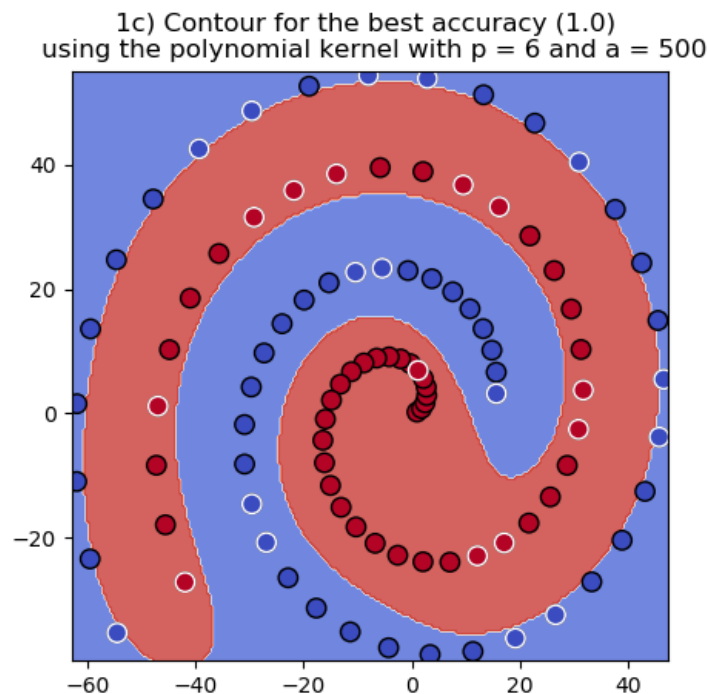
$$(x_1 < 0 \wedge x_2 > 0) \vee (x_1 > 0 \wedge x_2 < 0)$$

### 3.1

c) Determine experimentally, using the polynomial kernel, the value of  $p$  for which you get the best classifier (start with  $p = 1$ ) on your training data. Write down all the experiments performed, together with the classification error percentages and number of support vectors. Comment on the results you obtained.

Foi realizada a experiência para  $p$  entre 0 e 100 e para vários valores de  $a$ , de onde foi escolhido o valor 500. O parâmetro  $a$  corresponde ao "bias" do classificador e ao parâmetro `coef0` no código.

Com valor de `coef0` = 0, que corresponde ao valor padrão, a accuracy máxima era da ordem dos 60%. Mudando o valor de  $a$  para 500 conseguimos obter uma accuracy de 100%. Com o aumento do grau do polinômio ( $p$ ), estamos a aumentar a flexibilidade do classificador. É possível verificar que com este aumento da flexibilidade do classificador, a curvatura e a definição da separação também aumentam. Dos dados é possível verificar que aumentando o grau do polinômio, o valor do erro começa a diminuir até ao caso ideal, ponto a partir do qual volta a aumentar e estabiliza nos 50%. Já o número de vetores de suporte diminui à medida que se aumenta  $p$ . Verificámos que este grau do polinômio ideal é 6 ( $p=6$ ), onde obtivemos um número de vetores de suporte igual a 27 e um erro de classificação igual a 0. De salientar que o caso ideal não é só o que tem o erro igual a zero (como no caso de  $p=7$  até  $p=11$ ), mas também o menor número de vetores de suporte.

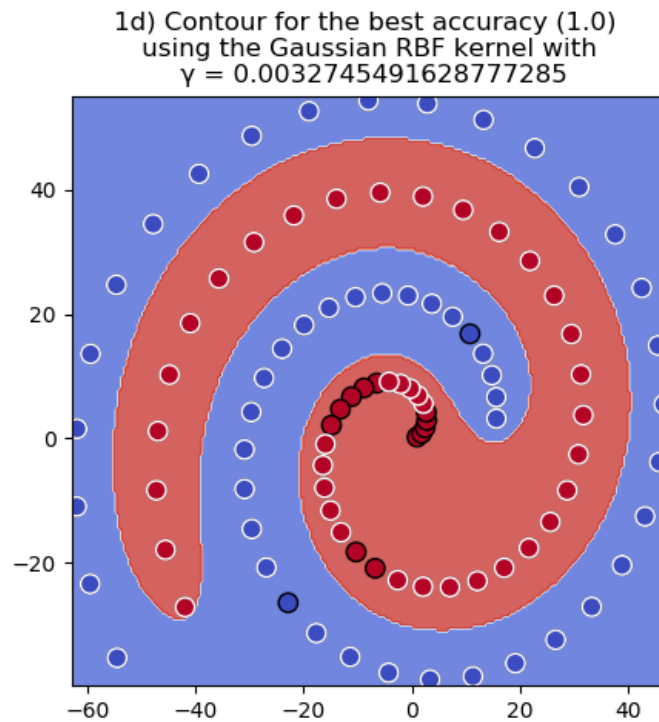


d) Using the same data try now the Gaussian RBF kernel. Find the approximate value of  $\gamma$  for which you can get the best classifier. Comment on the effect of  $\gamma$  on your results.

Foram realizados diversos testes, em que o valor de  $\gamma$  foi sendo mudado até se atingir a solução ótima. Como referido na alínea anterior, o classificador ótimo para uma situação é aquele que tem um menor valor de erro, mas também um menor valor de vetores de suporte. Este valor foi determinado como sendo  $\gamma = 0.0032745491628777285$ . Novamente, o número de vetores de suporte vai diminuindo, mas desta vez com a diminuição de  $\gamma$ , até que se atinge uma situação em que o número de vetores volta a aumentar.

Foi testado um valor de  $\gamma$  muito baixo (0.0001) e, como esperado, foi obtido "overfitting", sendo que quase todos os pontos do dataset são vetores de suporte (98).

Por outro lado, com um valor de  $\gamma$  muito elevado (a partir de 0.4328761281083057) a totalidade dos pontos (100) é utilizada como vetores de suporte, sendo que também ocorre "overfitting".



## 3.2

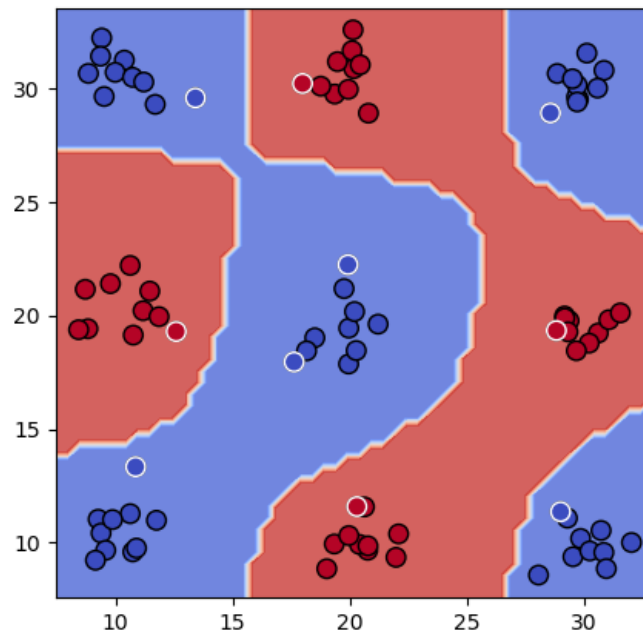
c) Find a value of  $\gamma$  that approximately minimizes the number of support vectors, while correctly classifying all patterns. Indicate the value of  $\gamma$  and the number of support vectors.

O valor de  $\gamma$  ótimo é aquele que reduz tanto o número de vetores de suporte como o erro. Neste caso, o valor obtido, que preserva a "accuracy" foi  $\gamma = 0.0011497569953977356$ . Este valor foi o escolhido uma vez que é o que proporciona menor erro (0%) e menor número de vetores de suporte (10), para percentagem de erro de 0%.

Obtém-se assim um hiperplano de decisão mais abrangente - menos curvatura.

À medida que se aumenta o valor de  $\gamma$ , a accuracy continua igual, isto é, com uma percentagem de erro de 0%, mas aumenta o número de vetores de suporte, o que fará com que ocorra "overfitting". Por exemplo, para gamma igual a 32.74, o número de pontos usados como vetores de suporte é 90, num total de 100 pontos classificados, o que é indesejável já que o classificador não generaliza suficientemente bem e irá provavelmente classificar incorretamente pontos que não pertençam ao conjunto de treino.

2c) Contour for the best accuracy (1.0) with minimum number of support vectors (10.0), using the Gaussian RBF kernel with  $\gamma = 0.0011497569953977356$



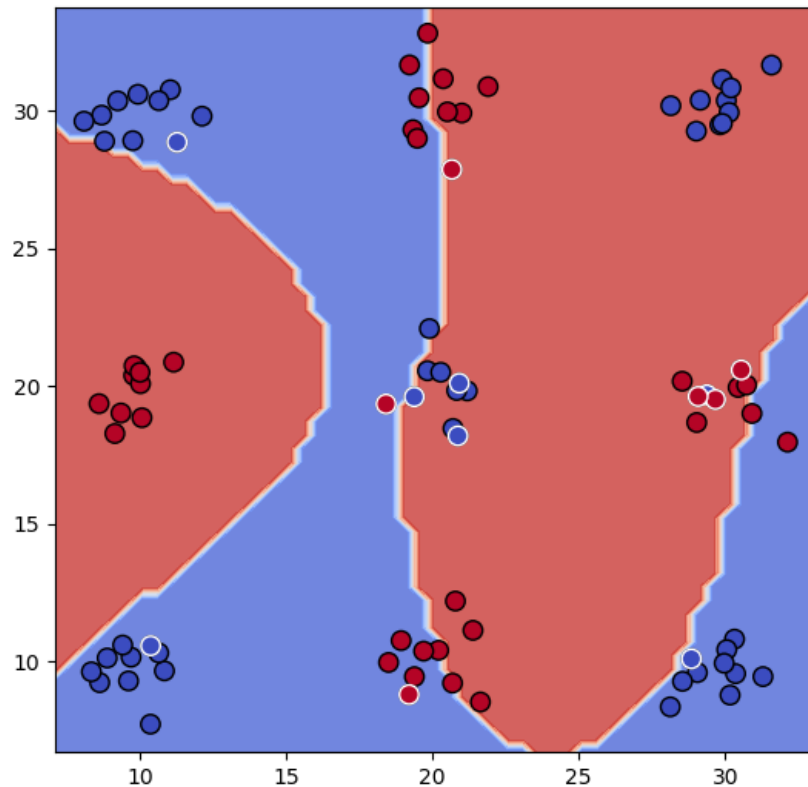
### 3.3

b) Run the classification algorithm on these data with the same value of  $\gamma$ , and comment on how the results changed, including the shape of the classification border, the margin size and the number of support vectors.

Em relação à alínea 3.2c, para o mesmo valor de  $\gamma$  utilizado, o número de vetores de suporte passou de 10 para 13 e a accuracy passou de 1 para 0.633, o que pode ser observado comparando os gráficos destas duas situações: no gráfico da alínea 3.2c, todos os pontos pertencem à região de decisão correta, mas no gráfico abaixo, existem pontos que estão na região de decisão errada. Isto resulta do facto de que, ao tentar encontrar fronteiras que incluam nas respetivas regiões de decisão todos os pontos, o algoritmo acaba por desenhar fronteiras que classificam erradamente muitos dos pontos.



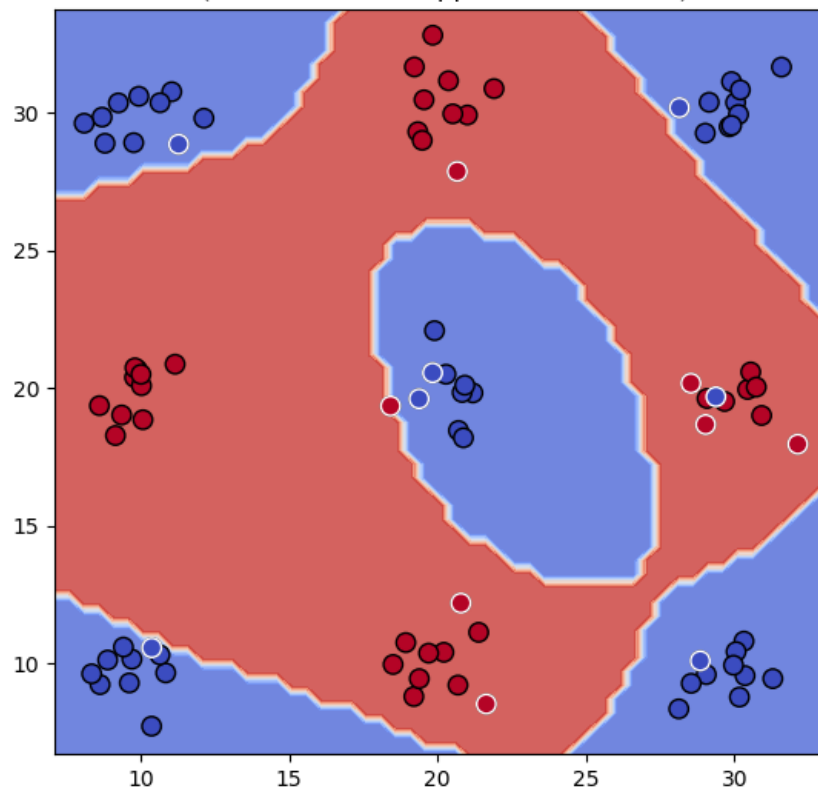
3b) Contour using the RBF kernel with  
 $\gamma = 0.0011497569953977356$ ,  
accuracy = 0.6333333333333333  
and number of support vectors = 13



c) Now reduce the value of 'C' parameter in order to obtain the so-called soft margin SVM. Try different values (suggestion: use powers of 10). Comment on the results.

Ao variarmos o valor de C, verificamos que, para valores muito pequenos, a accuracy é cerca de 0,55. À medida que se aumenta C, a accuracy vai aumentando até atingir um máximo de cerca de 0.99 para  $C \simeq 343046$ . A partir deste valor, quando se aumenta C, a accuracy começa a diminuir até 0,633 (situação da alínea 3.3b). Quanto ao número de vetores de suporte, embora existam algumas flutuações, podemos dizer que este número decresce à medida que se aumenta C. Na figura abaixo podemos ver as regiões de decisão para a situação em que C toma o valor que maximiza a accuracy.

3c) Contour using the RBF kernel with  
 $\gamma = 0.0011497569953977356$ , maximum accuracy(0.9888888888888889)  
for  $C = 343046.9286314926$   
(and number of support vectors = 14)



Como podemos ver, esta solução não é ideal, pois como tenta incluir um dos pontos vermelhos do conjunto central na fronteira de decisão vermelha, acaba por ter margens demasiado pequenas. Uma solução melhor seria a da figura abaixo que, apesar de ter menor accuracy, tem muito maiores margens.

3c) Contour using the RBF kernel with  
 $\gamma = 0.0011497569953977356$ , accuracy = 0.9777777777777777  
for  $C = 12328.467394420684$   
(and number of support vectors = 17)

