

Support Vector Machine

Gabriel Costa Leite

Introdução

O que é Support Vector Machine?

O Support Vector Machine ou Máquina de Vetores de Suporte, é um método de aprendizado supervisionado utilizado para classificar, criar um modelo de regressão e detectar anomalias em diversos conjuntos de dados.

Esse método se baseia no conceito de escolher um hiperplano que melhor divide o conjunto de dados, selecionando os pontos mais próximos de cada classe que maximizam a margem entre eles e o hiperplano escolhido. Esses pontos são chamados de Vetores Suporte.

Vantagens

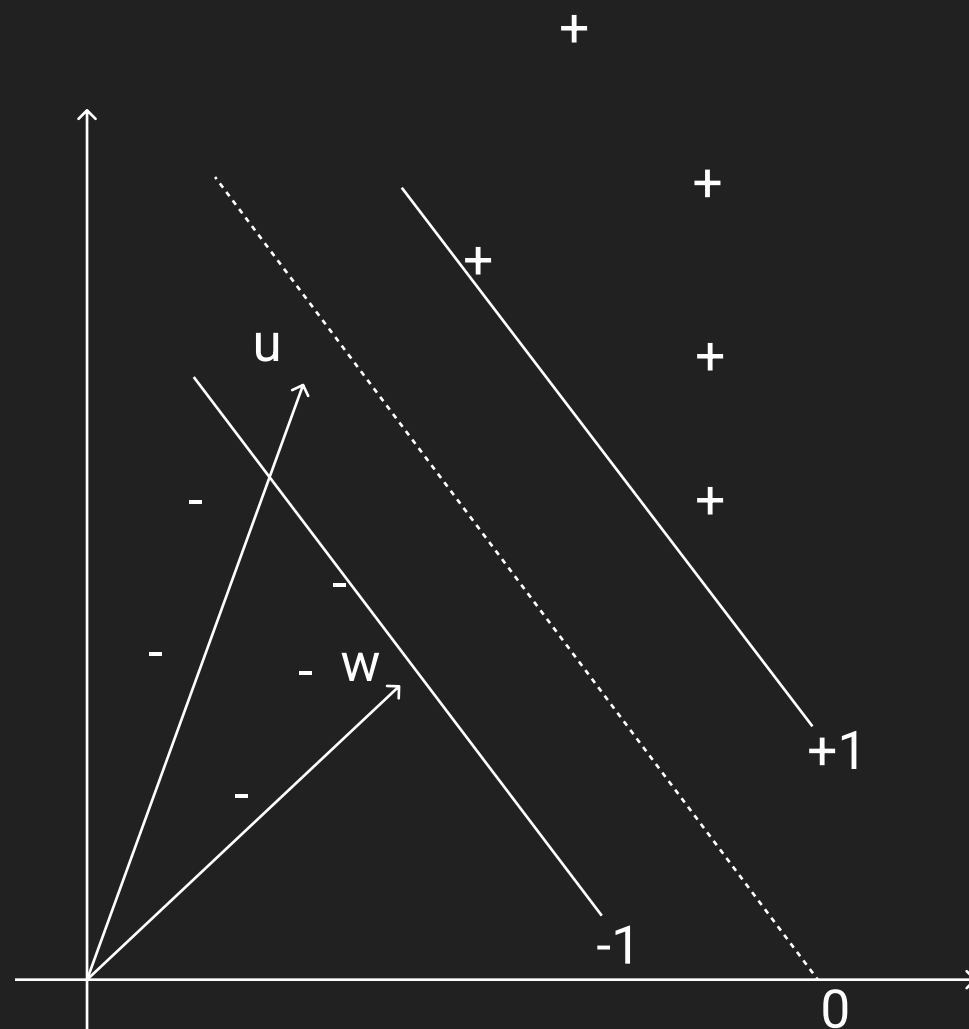
- > Eficaz em classificar conjunto de dados com diversas características.
- > Eficaz em classificar conjuntos de dados que possuem mais características do que amostras
- > Possui uma memória eficiente por conta dos vetores de suporte
- > É possível usar diversos tipos de Kernels, o que torna o método versátil

Desvantagens

- > É suscetível a problemas de sobreajustes (over fitting)
- > Possui um processo custoso de validação
- > É um classificador binário, ou seja, requer outras técnicas para classificar conjuntos de dados com mais de uma classe

Modelo Matemático

Função decisão



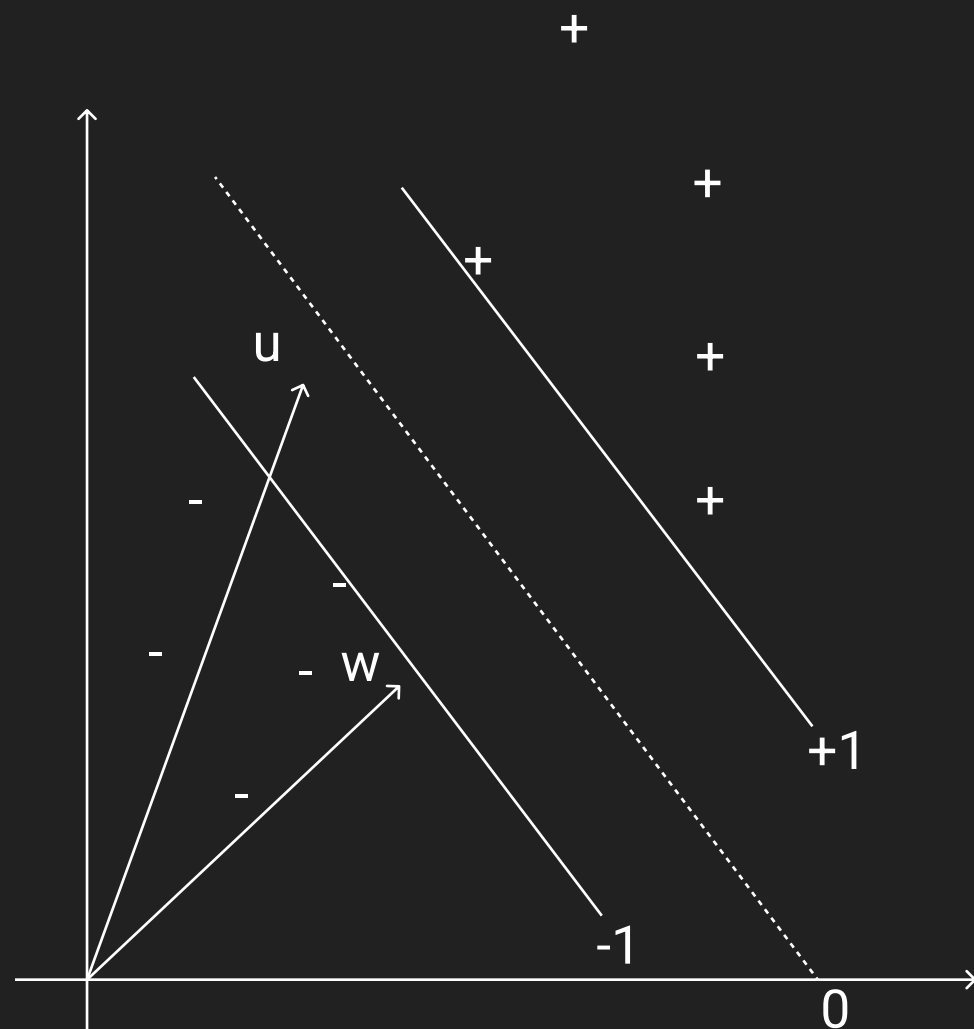
Decisão dentro da margem:

$$\vec{w} \cdot \vec{u} + b \geq 0 \rightarrow amostra +$$

Decisão fora da margem:

$$\begin{cases} \vec{w} \cdot \vec{x}_+ + b \geq 1 \\ \vec{w} \cdot \vec{x}_- + b \leq -1 \end{cases}$$

Função decisão



Classificação das amostras:

$$y_i = \begin{cases} +1 \rightarrow \text{sample } + \\ -1 \rightarrow \text{sample } - \end{cases}$$

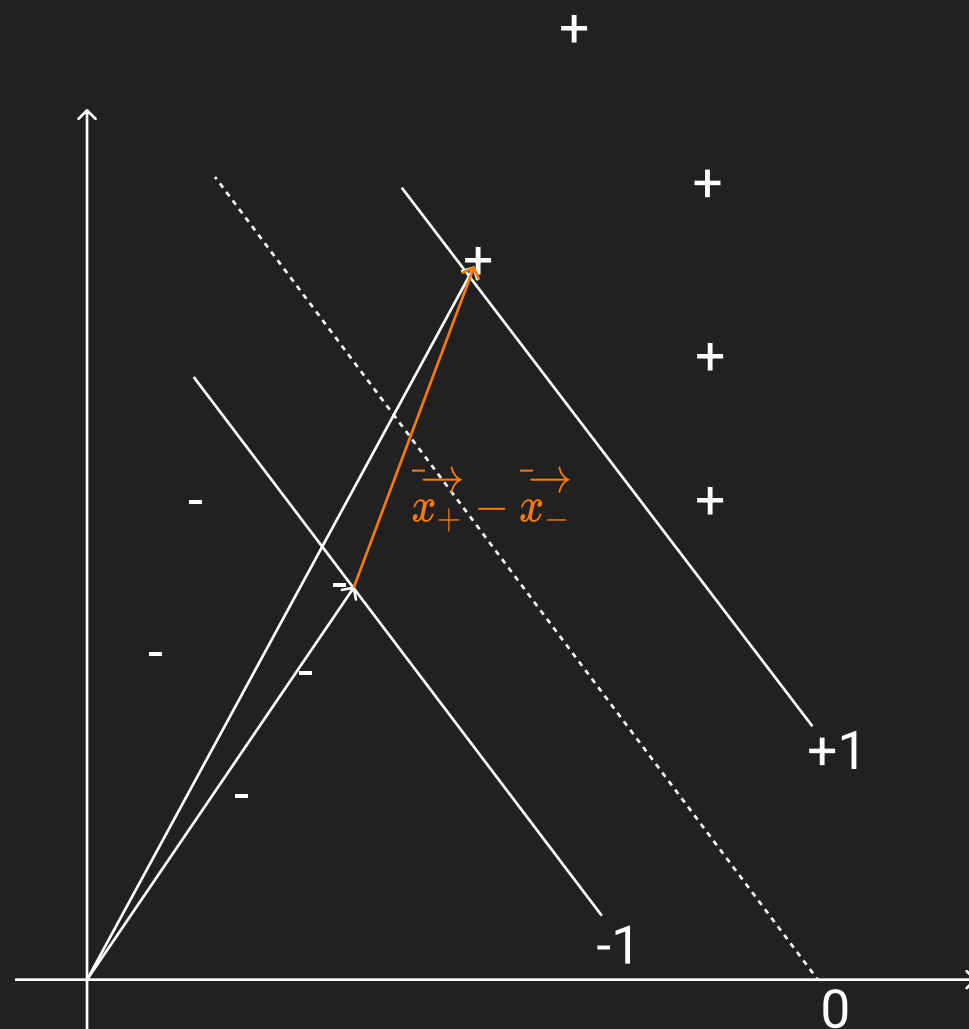
Classificação do modelo:

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0$$

Vetores de suporte:

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

Função decisão



Maximizar a distância:

$$d = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{\|\vec{w}\|}$$

$$d = (1 - b - (1 + b)) \cdot \frac{\vec{w}}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}$$

Ou seja,

$$\max\left(\frac{2}{\|\vec{w}\|}\right) \therefore \max\left(\frac{1}{\|\vec{w}\|}\right) \therefore \min(\|\vec{w}\|) \therefore \min\left(\frac{1}{2}\|\vec{w}\|^2\right)$$



Otimizando L:

Logo, a função de decisão será:

$$\sum \alpha_i y_i \vec{x}_i \cdot \vec{x}_j + b \geq 0 \rightarrow \text{sample} +$$

Kernel Trick

Kernel Trick:

$$K(x_i, x_j) = \overrightarrow{\phi(x_i)} \cdot \overrightarrow{\phi(x_j)}$$

Radial Basis Function:

$$K = e^{-\gamma(x_i - x_j)^2}$$

Função decisão:

$$\sum_{i \in SV} \alpha_i y_i K(x_i, x_j) + b$$

Problema de otimização

Quais os vetores de suporte?

$$\max_{\alpha} L = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$0 \leq \alpha_i \leq C$$

$$\sum \alpha_i y_i = 0$$

Problema de otimização!

Condições de KKT

Sequential Minimal Optimization

$$\alpha_i = 0 \rightarrow y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

$$\alpha_i = C \rightarrow y_i(\vec{w} \cdot \vec{x}_i + b) \leq -1$$

$$0 < \alpha_i < C \rightarrow y_i(\vec{w} \cdot \vec{x}_i + b) = -1$$

Condições de Karush-Kuhn-Tucker

Esse método também otimiza b !

Algoritmo

Algoritmos para SVM

O principal jeito de usar o SVM é utilizando a biblioteca de SVM do repositório scikit-learn, escrito em Python. Nesse software de código aberto é possível encontrar toda a documentação necessária para trabalhar com SVM, além de possuir exemplos e bancos de dados que podem ser utilizado para implementação de diversos algoritmo.

O ponto negativo de se utilizar essa biblioteca é não possuir acesso espontâneo ao otimizador e as funções matemáticas escritas em código.

Algoritmos para SVM

Com efeito, o autor dessa apresentação escreveu um código com o acesso ao otimizador e as funções matemáticas tendo como base toda matemática abordada anteriormente nessa apresentação.

Esse código e outras aplicações do SVM estão disponibilizadas no github:

https://github.com/gabrielclwq/Support-Vector-Machines/blob/main/svm_smo

Resultados

Iris Dataset do scikit-learn

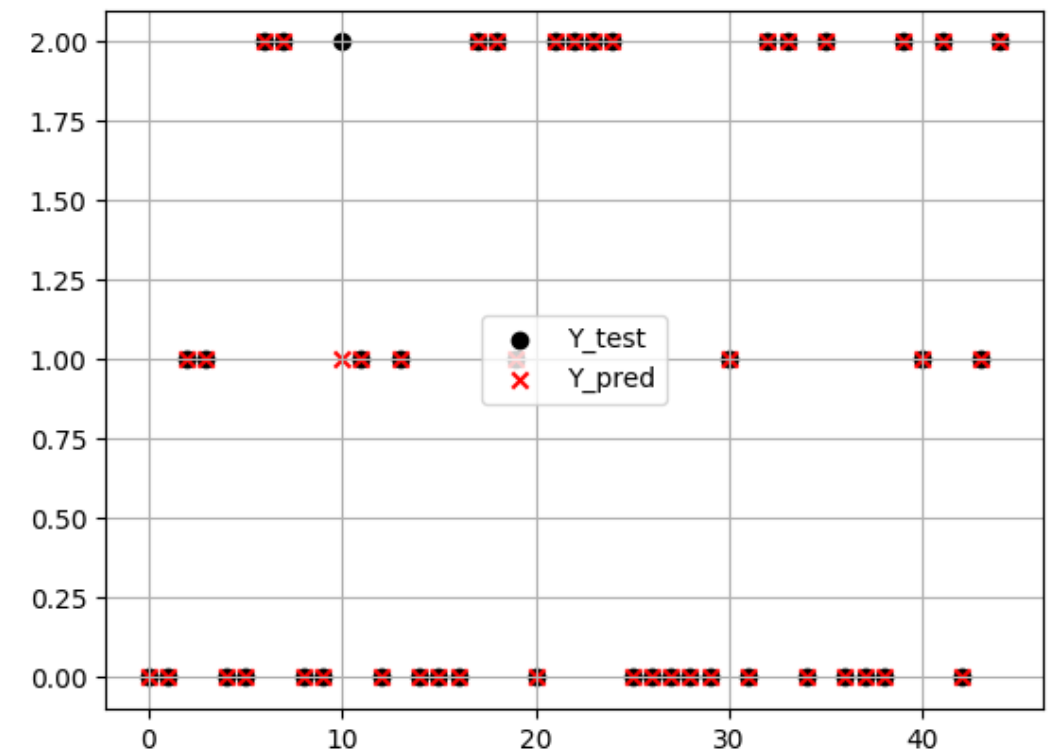
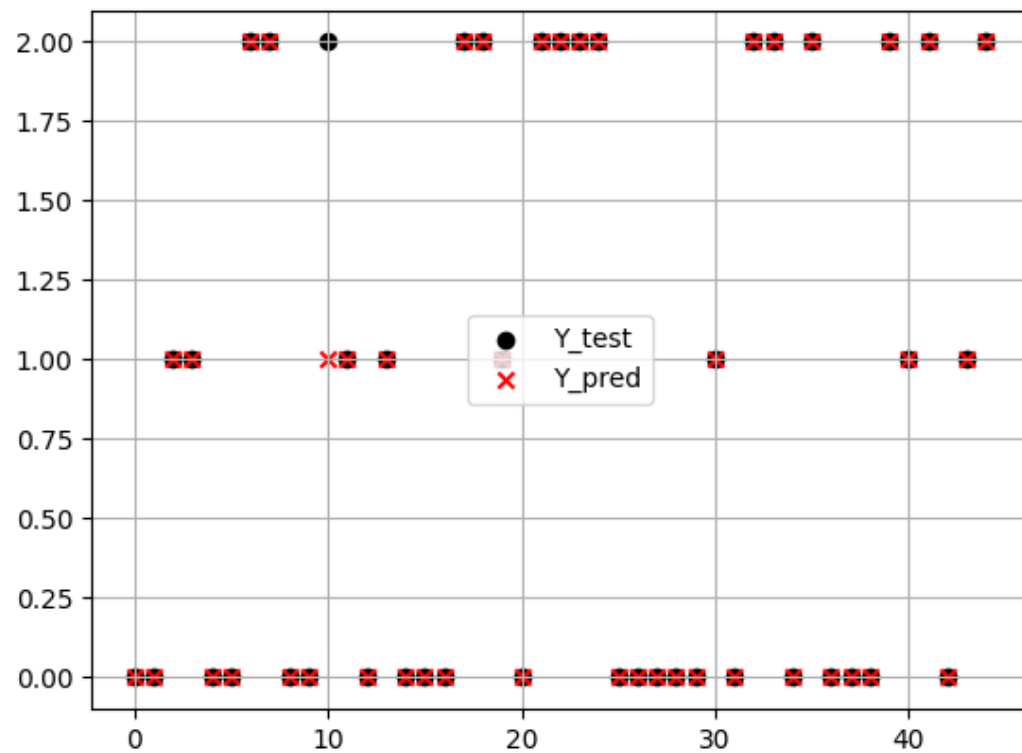
Esse dataset possui 150 amostras, onde as entradas são relacionadas ao comprimento da sépala, a largura da sépala, o comprimento da pétala e a largura da pétala, classificando flores do tipo íris em setosa, versicolor ou virginica.

Esse conjunto de dados foi dividido aleatoriamente em amostras de treinamento e amostras de teste utilizados para comparar os resultados obtidos para o algoritmo do autor e do sklearn.

Predição das classes

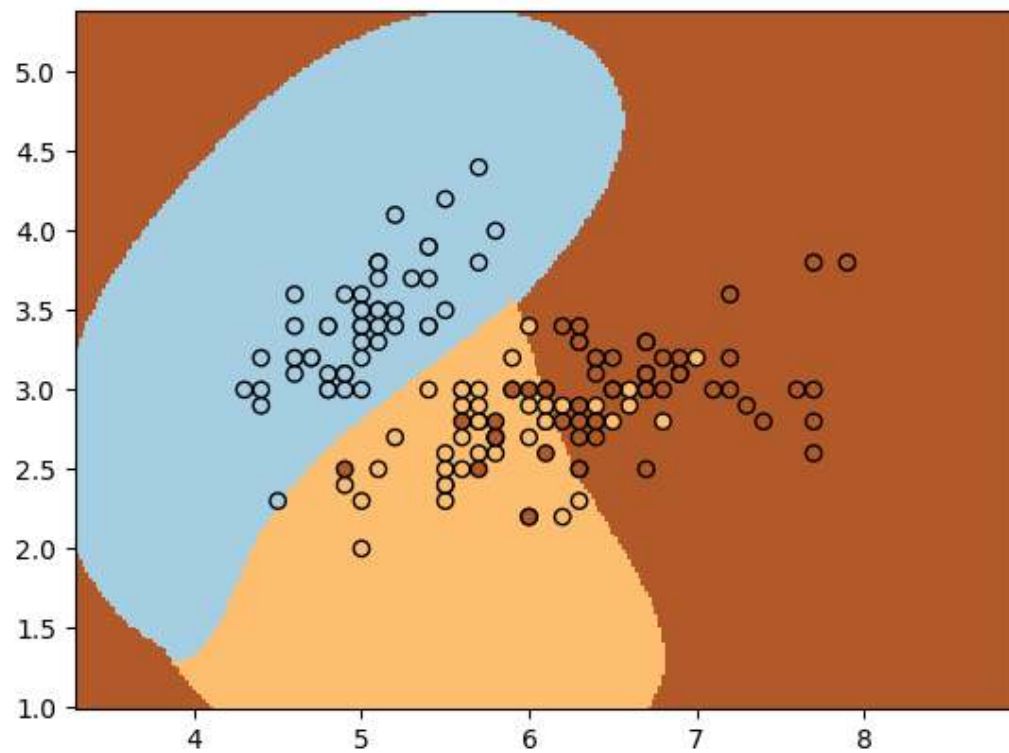
Autor:

Sklearn:

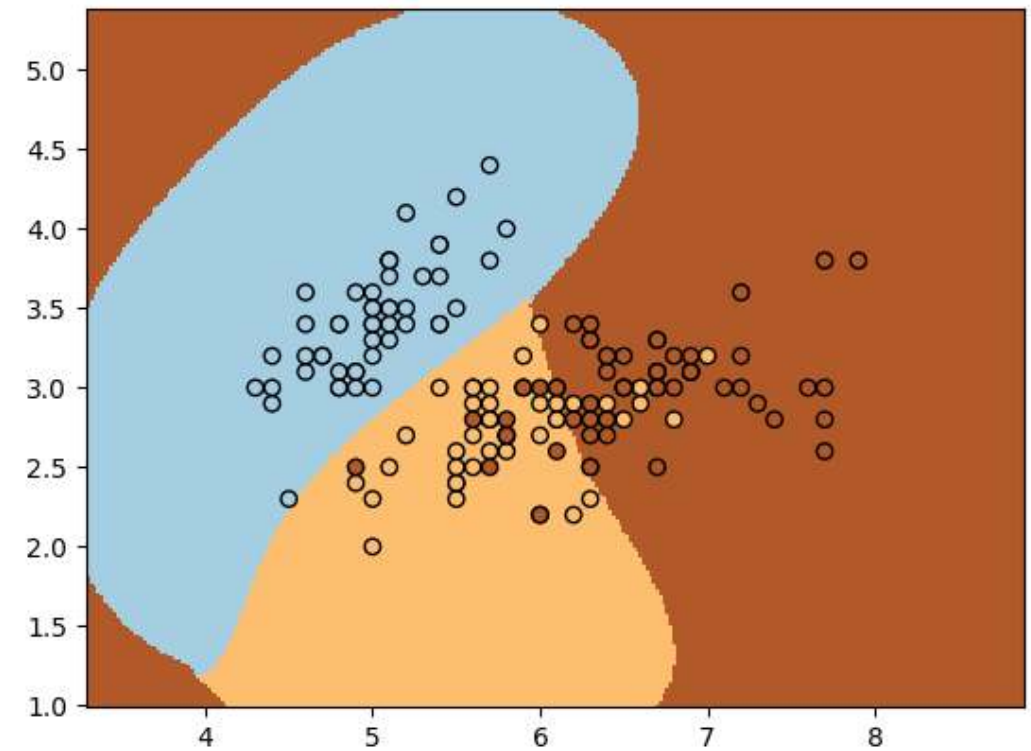


Decisão de superfície

Autor:



Sklearn:



Conclusão

Conclusão

Os resultados comparativos mostraram a eficiência do algoritmo do autor, onde é possível observar valores iguais nos dois testes. Entretanto, na compilação do código, o do autor demorou mais tempo do que o do sklearn.

Outrossim, é possível perceber que o coração do método do Support Vector Machine se encontra na otimização do alfa e do b, os quais são responsáveis por encontrar os melhores vetores suporte. Com efeito, o Kernel Trick é fundamental, uma vez que diminui o esforço computacional de operações matriciais e facilita a matemática do problema.

Portanto, o SVM é um método excelente para a classificação de dados.