* Envolve teorias como Learning Theory, Kernel Theory, Constrained Optimization Theory.
* Basicamente tem 3 componentes principais:
  + Um subconjunto do conjunto de treinamento, chamados **Support Vectors**
  + Pesos para cada um desses vetores
  + Uma função de similaridade, que retorna um valor entre -1 e 1, chamada **Kernel.**
* Pode-se dizer que o SVM aprende instâncias do conjunto de treinamento. Pois os Support Vectors são algumas instâncias do conjunto de treinamento. (**Instance Based Learning**)
* Também pode ser visto como uma generalização do Perceptron. No entanto tem 3 recursos cruciais adicionados: Kernels, Maximize Margin, variáveis de folga
* **Kernel Trick** é o termo técnico utilizado para designar a percepção que essa função pode ser qualquer função. Com isso, podemos encontrar fronteiras de decisão mais complicadas, do que apenas retas.
* O que uma função de kernel faz, continua sendo um produto cartesiano, no entanto em outro espaço, ou seja, faz uma transformação. E o problema continua sendo otimização dos pesos relacionados a cada vetor.
* Por exemplo, usando um Kernel Polinomial de grau 2, podemos representar fronteiras de decisão quadráticas! Ao invés de retas, podemos ter fronteiras como círculos, parábolas e hipérboles.