**1. Basic Concept**:

* **Decision Boundary**: In classification problems, we often want to find a decision boundary that separates different classes. For a two-class problem in a two-dimensional space, this boundary might be a line.
* **Support Vectors**: These are the data points that are closest to the decision boundary. The decision boundary (or hyperplane in higher dimensions) is determined in such a way that the margin between the two classes is maximized, and it's these "border" samples that define this margin.

**2. Linear vs Non-linear SVM**:

* **Linear SVM**: If data is linearly separable, SVM will find a linear decision boundary (a line in 2D, a plane in 3D, and a hyperplane in higher dimensions).
* **Non-linear SVM**: For non-linearly separable data, SVM can still be used by projecting the data into a higher-dimensional space where it becomes linearly separable. This is achieved using the kernel trick.

**3. The Kernel Trick**: A kernel is a function that computes the dot product of the vectors in some higher-dimensional space without explicitly having to compute the coordinates of the points in that space. Common kernels include:

* **Linear Kernel**:
* **Polynomial Kernel**:
* **Radial Basis Function (RBF) Kernel**:
* **Sigmoid Kernel**:

**4. Hyperparameters**:

* **C (Regularization Parameter)**: It determines the trade-off between achieving a low error on the training data and maximizing the margin. A smaller C creates a wider margin but may misclassify more training points, while a larger C creates a narrower margin but classifies more training points correctly.
* **Kernel Specific Parameters**: Parameters like degree in polynomial kernel, or gamma in RBF kernel, can affect the complexity and fit of the decision boundary.

**5. SVM for Regression (SVR)**: SVM can also be used for regression tasks, termed as Support Vector Regression (SVR). The main idea here is not to separate two classes but to fit a hyperplane such that most of the data lies within a certain margin or error rate.

**6. Advantages**:

* Effective in high dimensional spaces.
* Uses a subset of training points in the decision function (the support vectors), making it memory efficient.
* Versatile: Different kernel functions can be specified for the decision function.

**7. Disadvantages**:

* Doesn't provide probability estimates directly.
* Can be inefficient on very large datasets.
* Choosing the correct kernel and hyperparameters can be tricky.

**8. Practical Tips**:

* Always normalize data before using SVM.
* If features outnumber samples, avoid over-fitting by choosing a linear kernel.
* Use cross-validation to find the best hyperparameters.

SVMs are powerful tools in a data scientist's arsenal, but like all models, they are most effective when used with an understanding of the underlying data and problem context.