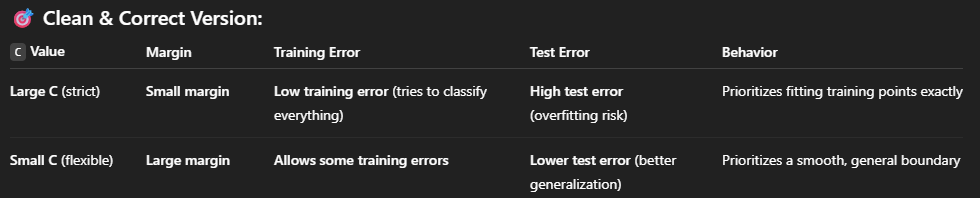
## **SVM:**

Large C(very strict) -> Small Margin -> Larger testing errors but smaller training errorsSmall C(not so strict) -> Large Margin -> Small testing errors a few more training errors



**Summary**

**Large C** makes the model **strict** — it **tries to avoid training errors** at all costs, even if that means having a **small, tight margin**, which may cause it to **overfit** and make more mistakes on new data.

**Small C** allows some **training mistakes**, in exchange for a **wider margin** and better **generalization** — which often leads to **fewer errors on test data**.

## **SVM Hyperparameter Selection**

**1. Start by understanding your data:**

* 📈 Is it likely **linearly separable** (clean, simple)?  
  → Try kernel='linear'
* 🔁 Does it seem **curvy or complex**?  
  → Use kernel='rbf' for flexible boundaries

**2. Choosing C: Regularization strength**

* 🎯 **High C** (e.g. 100, 1000) → Use when:
  + You want **low training error**
  + You **don’t want to misclassify** important data (e.g., fraud detection)
  + You **trust your labels**
* 🌊 **Low C** (e.g. 0.1, 1) → Use when:
  + Your data has **noise or label errors**
  + You prefer a **simpler, smoother** model
  + You care more about **generalization**

**3. Choosing gamma: For RBF kernel**

* 🧠 **High gamma** (e.g. 10, 100) → Use when:
  + You want the model to be **very sensitive** to small changes
  + You’re okay with **overfitting slightly** to the training data
* 🌐 **Low gamma** (e.g. 0.01, 0.1) → Use when:
  + You want the model to be **smoother**
  + You want to **avoid overfitting**

**4. How to actually find good values:**

* 🔄 Use **grid search** or **manual testing**
* Start with:

C = [0.1, 1, 10, 100, 1000]

gamma = [0.001, 0.01, 0.1, 1, 10]

* Evaluate using **cross-validation** to test generalization

**Reminder**

Always balance between **training accuracy** and **test/generalization performance**.  
A perfect model on training data doesn’t mean it’ll work well in the real world.

ADABOOST: It’s when we add weights for the misclassified points, so the classifiers in the coming models can focus on the misclassified samples

* **High Bias, Low Variance** models tend to underfit data, as they are not flexible. **Linear models** fall into this category of models.
* **High Variance, Low Bias** models tend to overfit data, as they are too flexible. **Decision trees** fall into this category of models.

Precision: "Out of all the points predicted to be positive, how many of them were actually positive?"

Recall: "Out of the points that are labeled positive, how many of them were correctly predicted as positive?"