

- Machine Learning
 - Regression (회귀) ↗
 - (Linear Regression)
 - (Logistic Regression (Binary classification))
 - (Multinomial classification)
 - Neural Network (신경망) : ↗ DNN, CNN ↗
 - KNN (K-Nearest Neighbor) ↗ 최근접 이웃
 - SVM (support Vector Machine) ↗ (Regression classification)
 - 선을 Deep Learning이 비해 살짝 떨어져도, 하지만 모델 자체가 가려워요 (속도 중요)

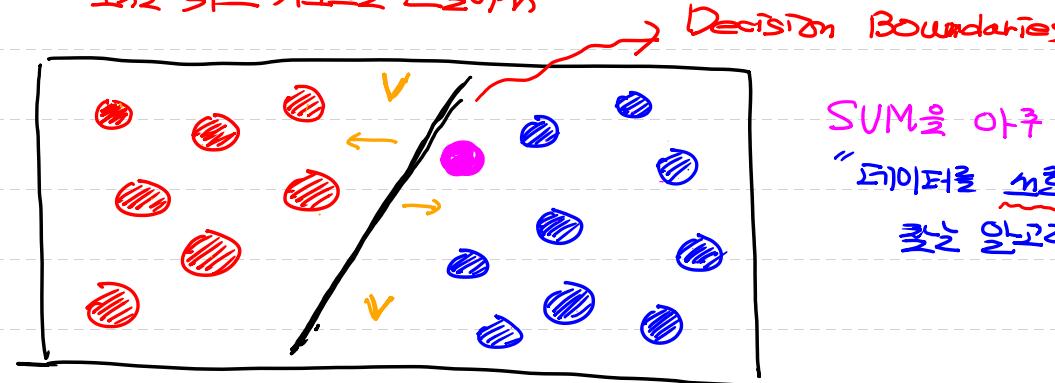
SVM

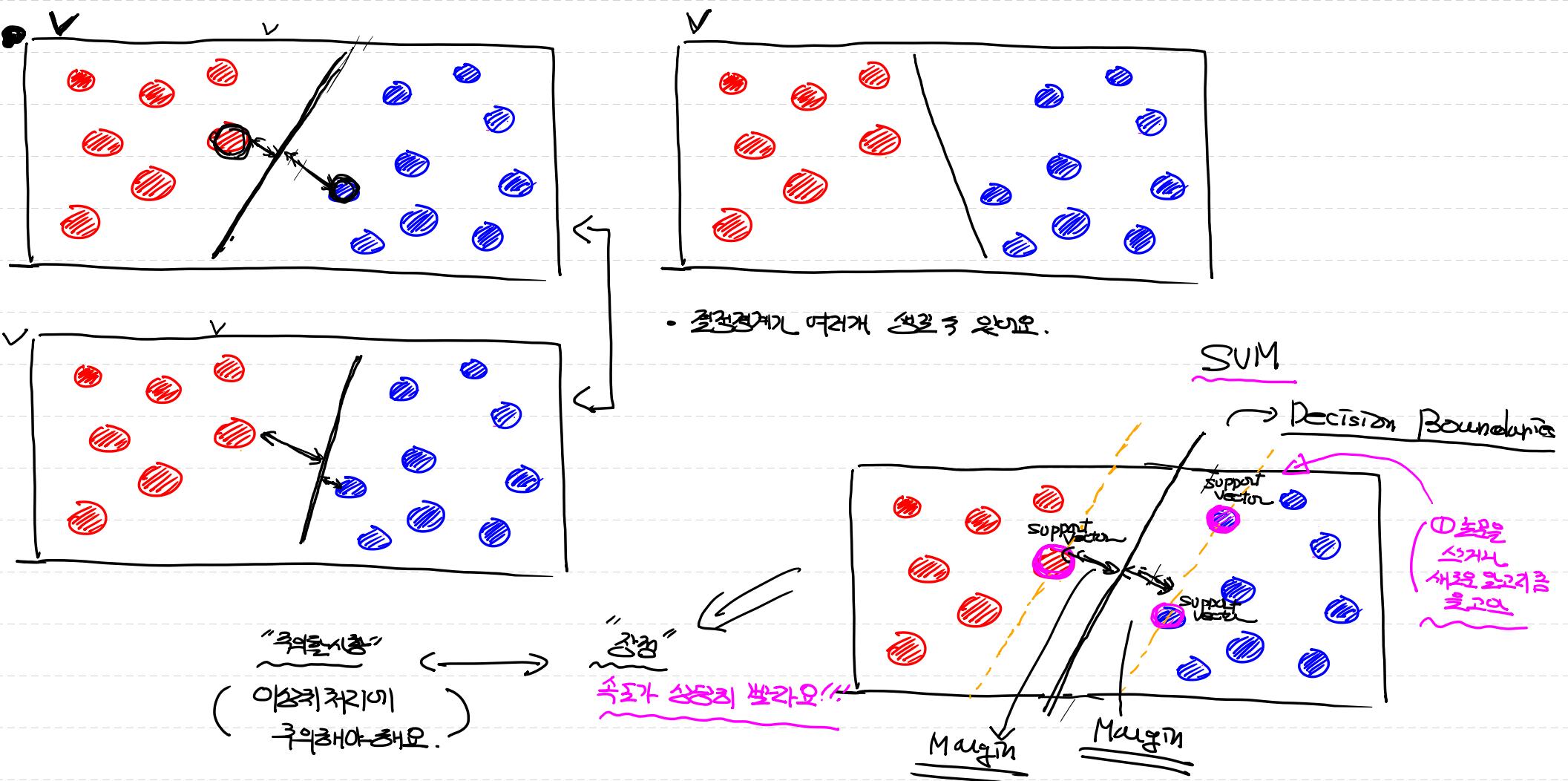
→ Decision Boundaries (결정경계)라는 개념을 이용해서

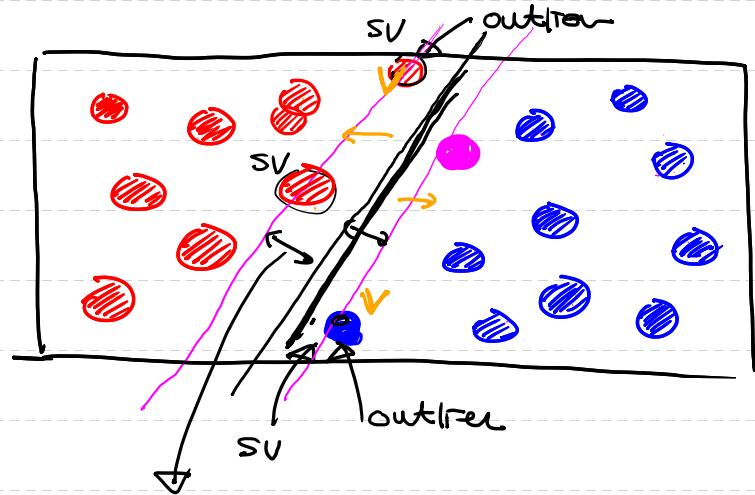
분류를 위한 기준선을 만들어서

SVM을 아주 간단하게 표현하면

“데이터를 선으로 분류하는 최적의 결정경계를 찾는 알고리즘” ↗ 직선을 의미하는게 아니예요

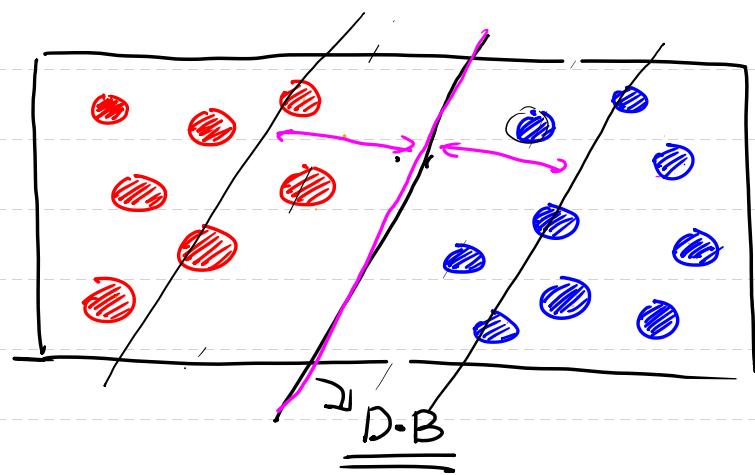






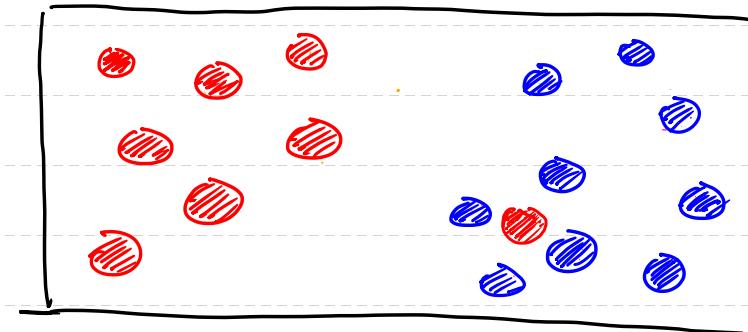
Margini가 작아지면 \Rightarrow 과대적합이 발생할
오류율 ↑

(hard margin)



$\underset{D \cdot B}{\equiv}$

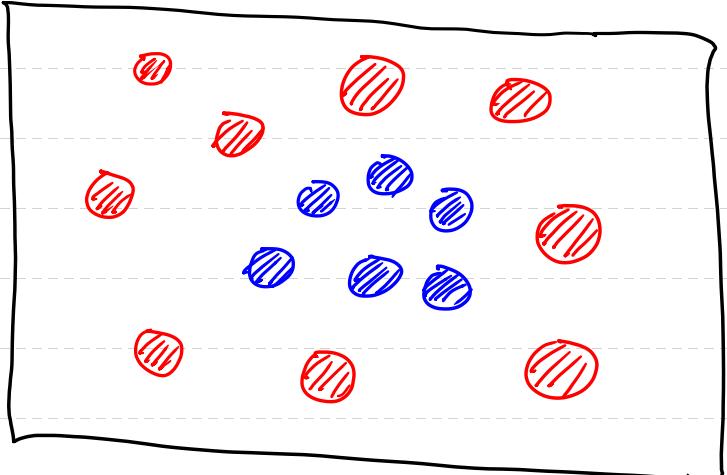
underfitting (과소적합)



➄ 데미터가 키의 그림처럼 초기화 경우 소형으로 데미터를
부여하기 힘들어도.
 \rightarrow 이를 제거해 해결하기 위해 데미터 오류를 감용하는
전략이 만들어져요 \Rightarrow Regularization

Sklearn의 $\underset{\text{cost}}{\equiv}$ "C" hyperparameter

'얼마나 많은 데미터
포인트가 다른 범주에 들어는 것을 감용할까'
기본값 $\underset{\text{C} \uparrow \rightarrow \text{과대적합}}{\equiv}$ "1"이고 이 값이 클수록 다른 범주에 들어는
데미터 포인트를 적게 감용



SVM

kernelartz hyper parameter

- "linear"
- "poly" 2차원을 3차원으로 사용 (polynomial)
- "rbf" (Radial basis function) 블라우저 층 층 "기본값"
상위 층의 파라미터로 사용 rbf kernel → 가우시안 커널

gammaartz
hyperparameter를 지정: → Decision Boundaries 를 얼마나
독립하게 그릴지에 대한 정도.
 (gamma ↓ D·B 칙칙에 가깝게
 ↑ D·B 구불구불하게)

- SUM model을 만드는 건 어려워요. ↗

하지만 Hyper parameter를 조절하며 최적화하는 작업은 어렵지 않아요.

→ 드로우하게 차리하면 → 최적의 model을 찾을 때까지 Hyper parameter를 수동으로 변경 실행

→ 이 과정을 자동화 시킬까 있을까???

Sklearn
Grid Search CV

Randomise Search CV

Cross Validation

Hyper parameter의 값을 몇 개
전래해요

범위를 지정하고 랜덤하게
추출해서 CV를 실행.