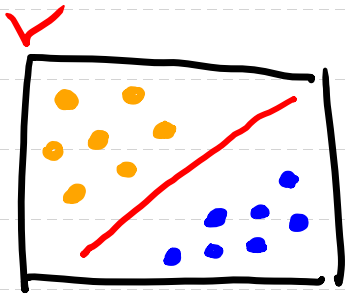
10/20 (화)

Sklearn으로 실습.

- SVM (support vector machine) [regression (얼마나) → SVR ✓ / classification (어떤것) → SVC ✓]

→ Deep Learning이 나오기로 인상이 좋긴 했지만 그래도 좋은 선생님이 모델자체도 가방끈을 중요.

SVM → Decision Boundaries (결정경계) ⇒ 데이터를 분류하기 위한 기준선.



새로운 데이터가 들어왔을때 기준선을 이용해서 어느 분류에 포함되는 가를 예측 [분류 (classification) / 분류안의 label의 평균 (regression)]

SVM → 데이터를 선형으로 분리하는 최적의 선형 결정경계를 찾는 알고리즘 !!

★ 선형은 직선을 포함하는 개념 (평면이나 곡선도 포함하는 개념)

직선 → 1차함수로 표현 ⇒ 선형은 1차함수로 표현되는 건가요?? ✗

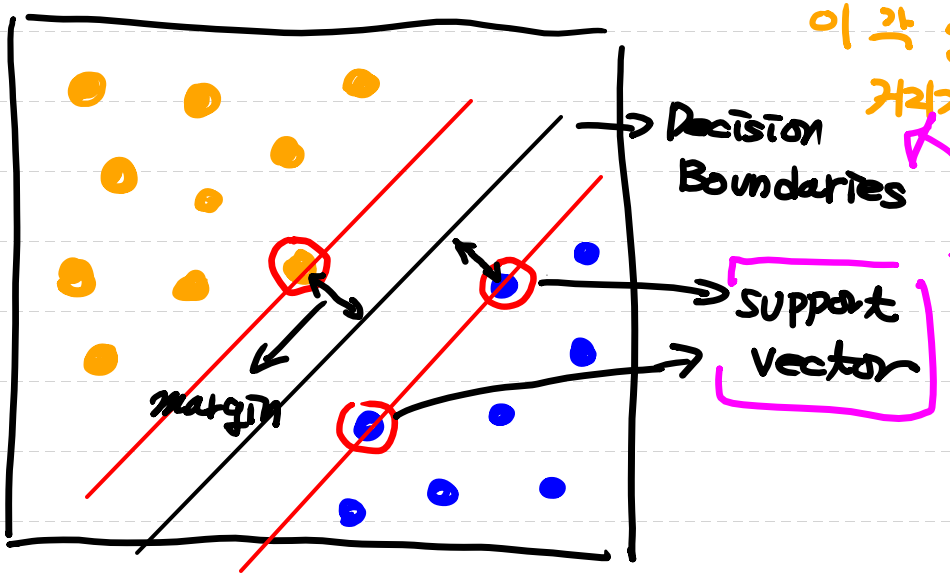
- feature 가 3개이면 Decision Boundaries 는 평면으로 표현

feature의 개수가 증가하면 Decision Boundaries 도 "고차원" → 결국 초평면 ∨ hyperplane ∨

Decision Boundaries 를 구하기 위해 ⇒ ∨ Support vector

↳ 각 영역에 들어있는 데이터 포인트

이 각 group의 support vector 들과의 거리가 가장 먼 support vector를 구해요!!



장점은 → 위의 support vector 만 이용하는 기법이기 때문에

⇓

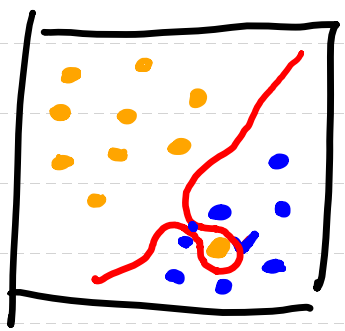
∨ 시간을 줄일 수 있어요!!

"C" 절충해줘야 하는값!!

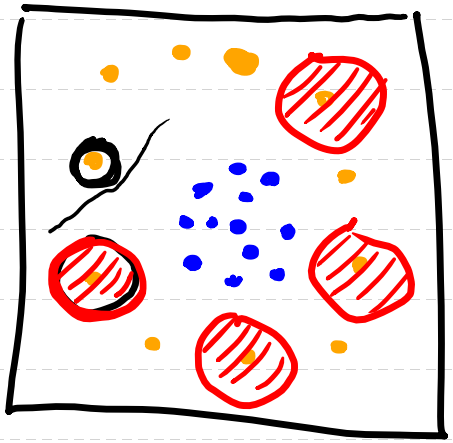
· 주의점 → 만약에 이상치(outlier)가 존재한다면.
이 이상치가 support vector가 될 수 있고
이런 경우 overfitting의 문제가 발생 ⇒ margin이 작아지게 되면
고차원적
⇓
Hard margin

⇒ 아니 그러면 이상치를 제거, 변경하면 되는거 아냐??
⇒ 현실 그런건 아니예요!

그렇기 때문에 모델을 만들때 → 약간의 오차를 허용하는 방식을 고려.



↓ sklearn에서는 cost라고 표현되는 "C" hyperparameter 이용
"C" 기본값 1
C의 값을 크게하면. 다른 범위에 있는 데이터 포인트를 조금 허용.
C의 값은 ↓
→ overfitting



→ "kernel 기법"

→ 주어진 data를 고차원의 공간으로 "projection" (투영)

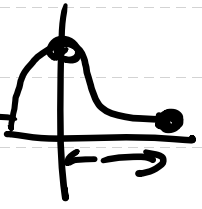
Sklearn [kernel 초기값 [linear (직선으로 Decision Boundaries 가 구해짐!!)
poly (2→3)
rbf (기본값) →

"gamma"

→ 데이터들이 영향력을 행사하는 거리

→ gamma 값이 클수록 영향력 행사 거리 ↓

gamma 값이 작을수록 " " " ↑



"gamma" hyperparameter.

→ Decision Boundaries를 보다 유연하게 그릴지를 조절하는 값!!

이 값이 크면 구불구불 → overfitting

" 작으면 직선. → underfitting

구불구불 → overfitting

· SVM ["C", "gamma"] ⇒ 잘 설정할 수 있을까요??

"hyperparameter"

① 수동으로 적절한 "C", "gamma" 값을 찾을거예요!! → X (노동집약적)

② Sklearn 최적의 parameter를 찾을 수 있는 방법을 제공 (자동화)

→ GridSearchCV

"Cross validation"

⇒ 사용할 parameter를 여러개 지정해 놓고 CV를 실행해서 최적의 accuracy를 도출하는 hyperparameter를 찾아줘요!

#코드를 가지고 알아보아요!