나이브 베이즈(Naive Bayes)

나이브 베이즈(Naive Bayes)

- 연속적인 특성으로 분류기 훈련
  - 베이즈 이론은 새로운 정보 P(A|B)와 사건의 사전 확률P(A)가 주어졌을 때 어떤 사건이 일어날 확률을 이해하는 방법입니다.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

```
from sklearn import datasets
from sklearn.naive_bayes import GaussianNB

iris = datasets.load_iris()  # 데이터 로드
features = iris.data
target = iris.target

classifer = GaussianNB()  # 가우시안 나이브 베이지 객체 생성
model = classifer.fit(features, target)  # 모델 훈련
new_observation = [[ 4,  4,  4,  0.4]]    #New Sample Data
model.predict(new_observation)   # 클래스 예측

# 각 클래스별 사전 확률을 지정한 가우시안 나이브 베이즈 객체 생성
clf = GaussianNB(priors=[0.25, 0.25, 0.5])
model = classifer.fit(features, target)  # 모델  훈련
```

나이브 베이즈(Naive Bayes)

- 연속적인 특성으로 분류기 훈련
  - 머신러닝에서는 베이즈 이론을 분류에 적용한 것이 나이브 베이즈 분류기입니다.
  - 나이브 베이즈 분류기 제공 기능

1. 직관적인 방법을 사용합니다.
2. 작은 양의 데이터에서 사용할 수 있습니다.
3. 훈련과 예측에 계산 비용이 적게 듭니다.
4. 환경이 바뀌더라도 자주 안정적인 결과를 만듭니다.

$$P(y \mid x_1, \cdots, x_j) = \frac{P(x_1, \cdots x_j \mid y) P(y)}{P(x_1, \cdots, x_j)}$$

- 나이브 베이즈 분류기는 데이터에 있는 각 특성에 대해 가능도의 통계적 분포 P(Xj | Y)를 가정해야 합니다.
- 정규분포(가우시안 분포), 다항 분포, 베르누이 분포를 자주 사용합니다.
- 특성의 성질(연속, 이진 등)에 따라 분포를 선택합니다.
- 나이브 베이즈 분류기는 각 특성과 특성의 가능도가 독립적이라고 가정합니다.

나이브 베이즈(Naive Bayes)

- 이산적인 카운트 특성으로 분류기 훈련
  - 다항 나이브 베이즈가 가장 많이 사용되는 경우 중 하나는 BoW(bag of words)나 tf-idf 방식을 사용한 텍스트 분류입니다.

```
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer

text_data = np.array(['I love Brazil. Brazil!',
                      'Brazil is best',
                      'Germany beats both'])

count = CountVectorizer()   # BoW(bag of words)를 만듭니다.
bag_of_words = count.fit_transform(text_data)
features = bag_of_words.toarray()   # 특성 행렬을 만듭니다.
target = np.array([0,0,1])  # 타깃 벡터를 만듭니다.
#MultinomialNB를 사용해 두 클래스(brazil과 germany)에 대한 사전 확률을 지정하여 모델을 훈련
# 각 클래스별 사전 확률을 지정한 다항 나이브 베이즈 객체를 만듭니다.
classifer = MultinomialNB(class_prior=[0.25, 0.5])
model = classifer.fit(features, target)  # 모델 훈련
new_observation = [[0, 0, 0, 1, 0, 1, 0]]  #New Sample Data
model.predict(new_observation)   # 새로운 샘플의 클래스 예측
```

나이브 베이즈(Naive Bayes)

- 이진 특성으로 나이브 베이지 분류기 훈련
  - 베르누이 나이브 베이즈 분류기는 모든 특성이 두 종류의 값만 발생할 수 있는 이진 특성이라고 가정합니다.
  - 텍스트 분류에 많이 사용됩니다.
  - 평탄화 매개변수인 alpha를 가지고 있고 모델 선택 기법을 사용해 튜닝해야 합니다.
  - 사전 확률을 지정하려면 class_prior 매개변수에 클래스별 사전 확률을 담은 리스트를 전달합니다.
  - 균등분포를 사용하려면 fit_prior=False로 지정합니다.

```
import numpy as np
from sklearn.naive_bayes import BernoulliNB

# 세 개의 이진 특성을 만듭니다.
features = np.random.randint(2, size=(100, 3))
# 이진 타깃 벡터를 만듭니다.
target = np.random.randint(2, size=(100, 1)).ravel()
# 각 클래스별 사전 확률을 지정하여 베르누이 나이브 베이즈 객체를 만듭니다.
classifer = BernoulliNB(class_prior=[0.25, 0.5])
model = classifer.fit(features, target)  # 모델  훈련

model_uniform_prior = BernoulliNB(class_prior=None, fit_prior=True)
```

나이브 베이즈(Naive Bayes)

- 예측 확률 보정
  - 나이브 베이즈에서는 타깃 클래스에 대한 예측 확률의 순위는 유효하지만 예측 확률이 0 또는 1에 극단적으로 가까워지는 경향이 있습니다.
  - 의미 있는 예측 확률을 얻으려면 보정이라 부르는 작업을 수행해야 합니다.
  - 사이킷런에서 CalibratedClassifierCV 클래스를 사용하여 잘 보정된 예측 확률을 k-폴드 교차검증으로 만들 수 있습니다.
  - CalibratedClassifierCV에서 훈련 세트를 사용해 모델을 훈련하고 테스트 세트를 사용해 예측 확률을 보정합니다.
  - 반환된 예측 확률은 k-폴드의 평균입니다. .

```
from sklearn import datasets
from sklearn.naive_bayes import GaussianNB
from sklearn.calibration import CalibratedClassifierCV

iris = datasets.load_iris()  # 데이터 로드
features = iris.data
target = iris.target

classifer = GaussianNB()  # 가우시안 나이브 베이즈 객체 생성
# 시그모이드 보정을 사용해 보정 교차 검증을 만듭니다.
classifer_sigmoid = CalibratedClassifierCV(classifer, cv=2, method='sigmoid')
classifer_sigmoid.fit(features, target) # 확률을 보정
new_observation = [[ 2.6,  2.6,  2.6,  0.4]]  #New Sample Data
classifer_sigmoid.predict_proba(new_observation)  # 보정된 확률을 확인
```

나이브 베이즈(Naive Bayes)

- 예측 확률 보정
  - CalibratedClassifierCV는 method 매개변수에서 두 개의 보정 방법을 지원합니다.
  - 플랫의 시그모이드 모델과 등위회귀입니다.
  - 등위회귀는 비모수 모델이기 때문에 샘플 크기가 작으면(예: 100개의 샘플) 과대적합되는 경향이 있습니다.

```
# 가우시안 나이브 베이즈를 훈련하고 클래스 확률을 예측합니다.
classifer.fit(features, target).predict_proba(new_observation)
classifer_sigmoid.predict_proba(new_observation) # 보정된 확률을 확인
```