한국기술교육대학교 능력개발교육원

과정구분
(교직훈련과정 등)

6 머신러닝 이해하기 - 이론과 실습

KOREATECH 한국기술교육대학교

KOREATECH 능력개발교육원

CONTENTS

KOREATECH 한국기술교육대학교

KOREATECH HRDI 능력개발교육원

01 학습, 모델, 예측

01-1 소단원명

KOREATECH 한국기술교육대학교

KOREATECH 능력개발교육원

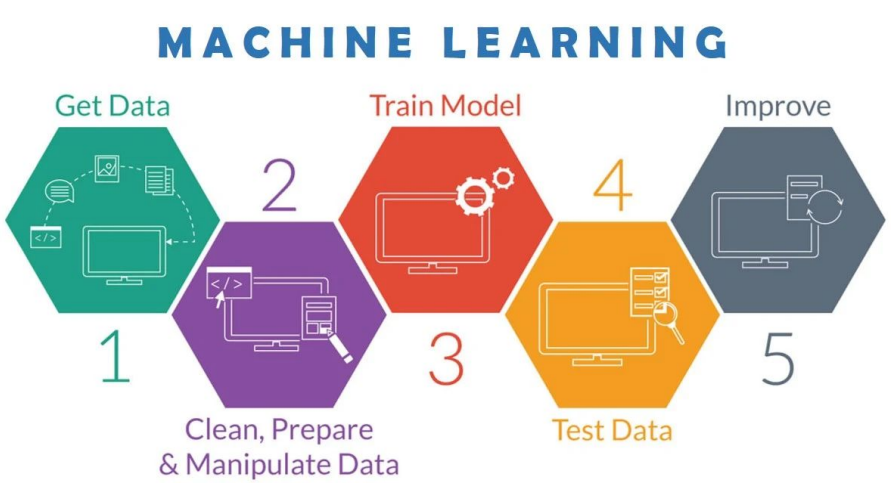
모델

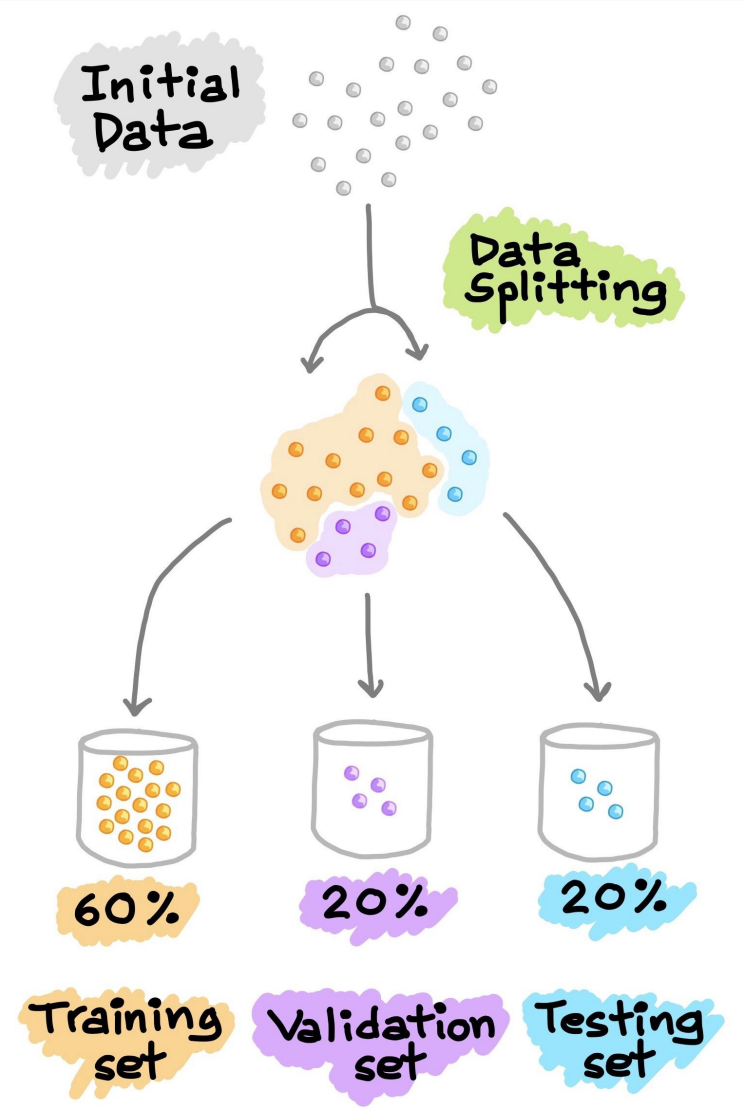
모델은 데이터와 밀접한 관계

어떤 모델을 사용하는가는 어떤 데이터를 사용하는가 중요한 영향을 미침

학습 > 테스트 -> 수정/개선 -> 다시 학습...

- Regression
- Classification
- Clustering
- Dimensionality Reduction
- Neural Networks
- Ensemble Methods
- Deep Learning





모델 학습

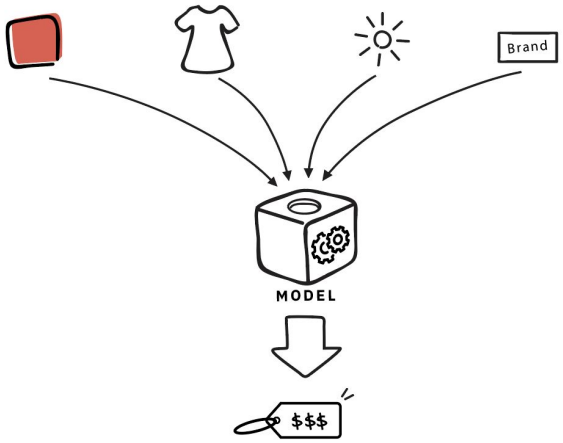
학습(training)도 중요하지만, 제대로 모델이 만들어졌는지 확인하기 위해서는 테스트(test)도 중요합니다.

data를 분리하는 방법중 하나로는:

- training set
- validation set
- testing set

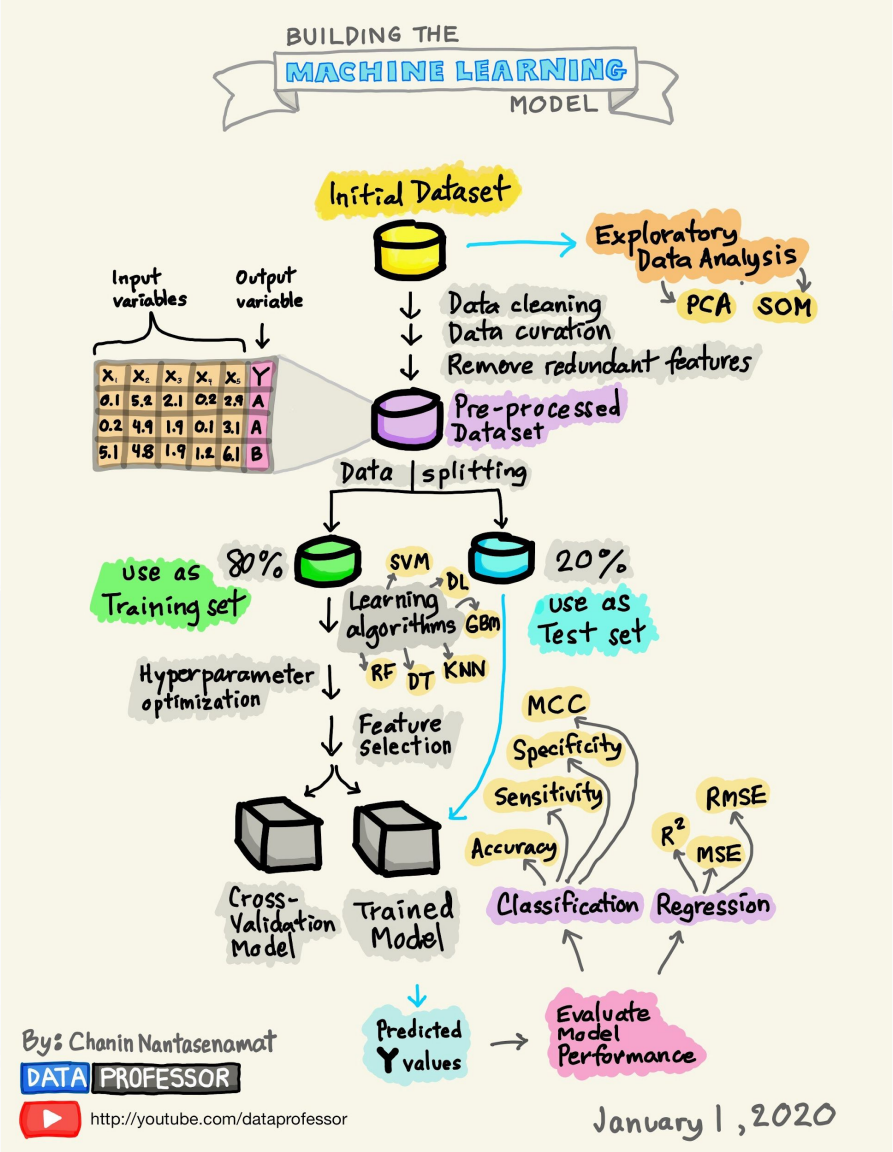
예측 (prediction)

- accuracy (정확도)
  - 정확도는 모델이 올바른 예측(correct prediction)을 하는 전체적인 빈도
- error rate (오류율)
  - 오류율은 전체적으로 모델이 잘못된 예측(wrong prediction)을 하는 빈도
- *recall, **precision 등등



*Recall is a measure that tells us how great our model is when all the actual values are positive.

**Precision is when the model predicts a positive value then what are the odds that the model has made a correct prediction.



Building the Machine Learning model

크게보기

https://towardsdatascience.com/how-to-build-a-machine-learning-model-439ab8fb3fb1

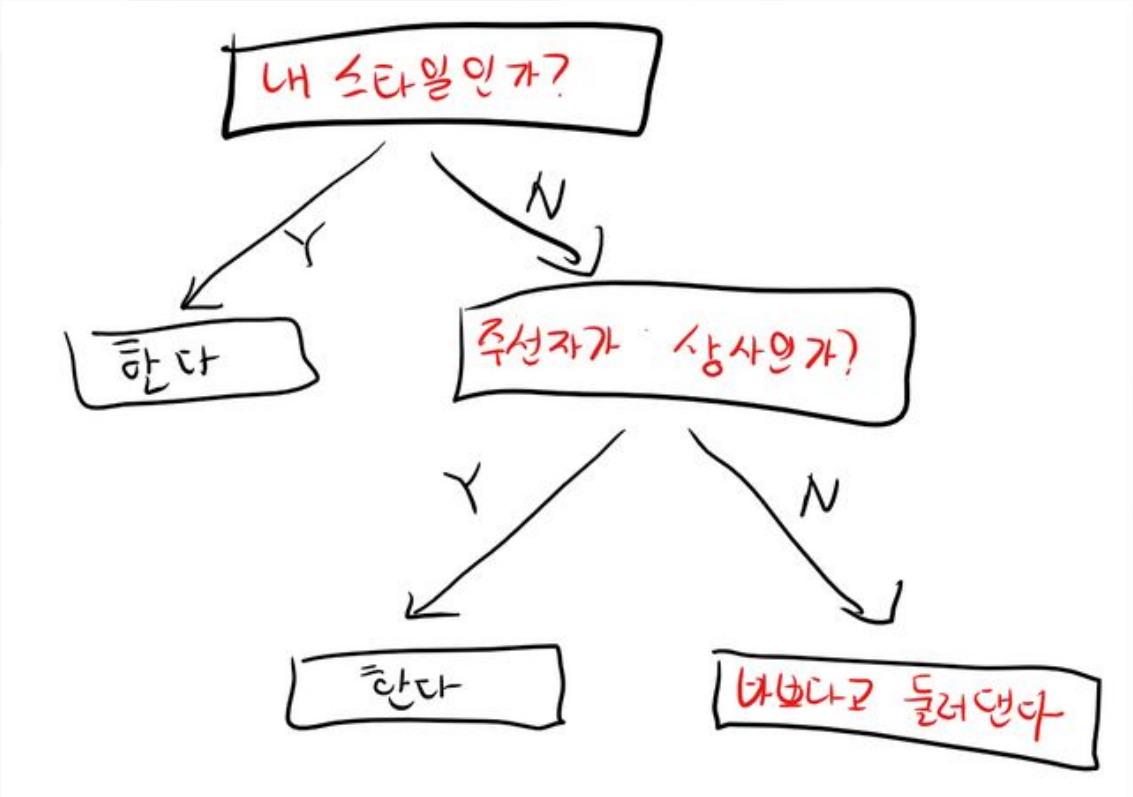
02 결정 트리 (Decision Tree)

KOREATECH 한국기술교육대학교

KOREATECH 능력개발교육원

결정 트리 (Decision Tree)

Decision Tree = 결정트리, 의사결정트리, 의사결정나무, 결정나무



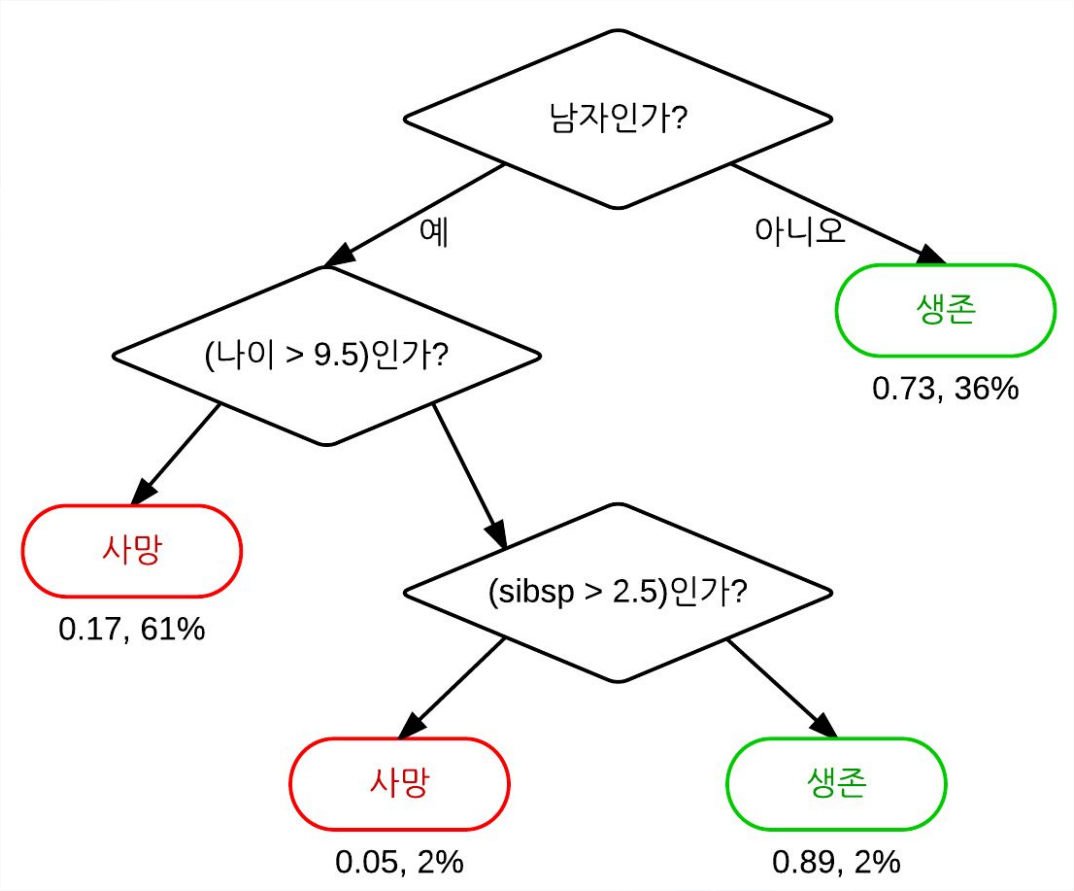
결정 트리 (Decision Tree)

타이타닉호 탑승객의 생존 여부를 나타내는 결정 트리.

("sibsp"는 탑승한 배우자와 자녀의 수를 의미한다.)

잎 아래의 숫자는 각각 생존 확률과 탑승객이 그 잎에 해당될 확률을 의미한다.

[출처: ko.wikipedia.org]



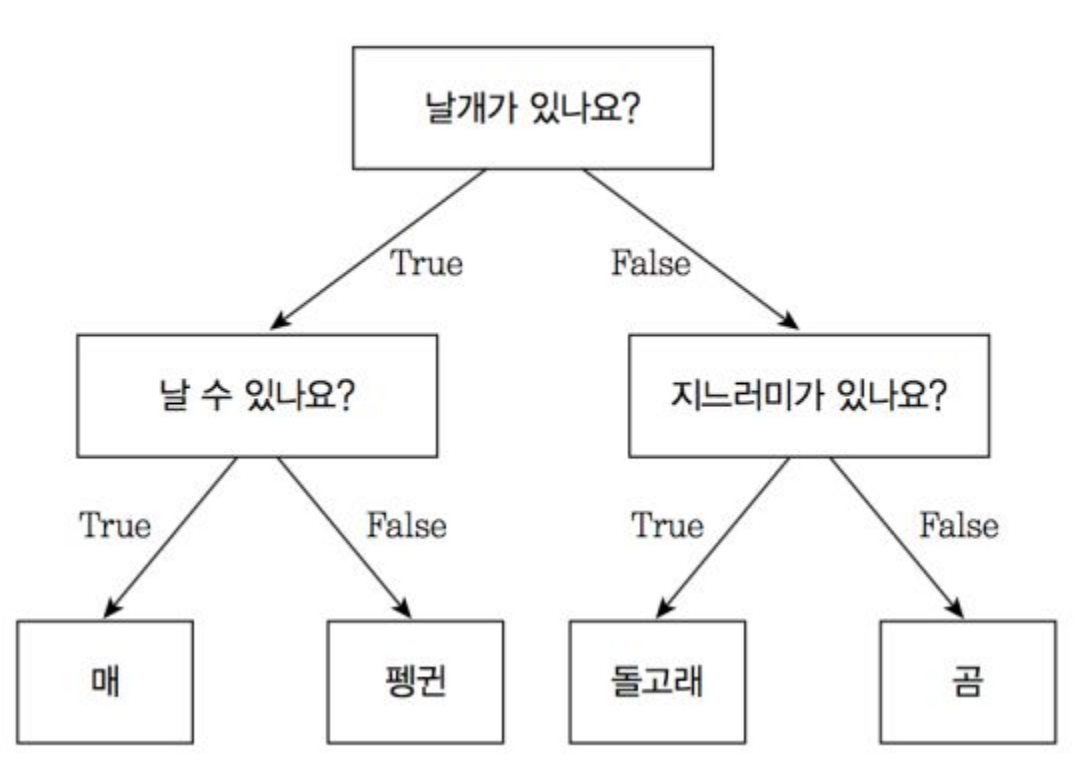
의사결정트리 (Decision Tree)

- 전형적인 분류 모델
- 매우 직관적인 방법 중 하나
- 다른 모델들과 다르게 결과물이 시각적으로 읽히기 쉬운형태가 장점

- 어떤 곳에 사용되나요?
  - 대출을 원하는 사람의 신용평가
  - 독버섯과 버섯을 분류
  - 실질적인 분류에 자주 사용됨

정보획득량 / 엔트로피의 개념 이해가 필요.

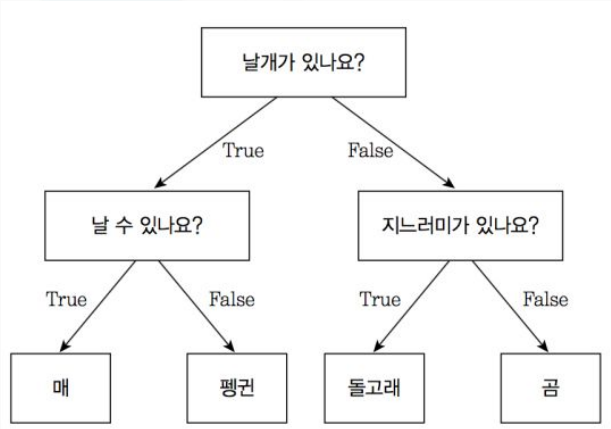
결정 트리 (Decision Tree)



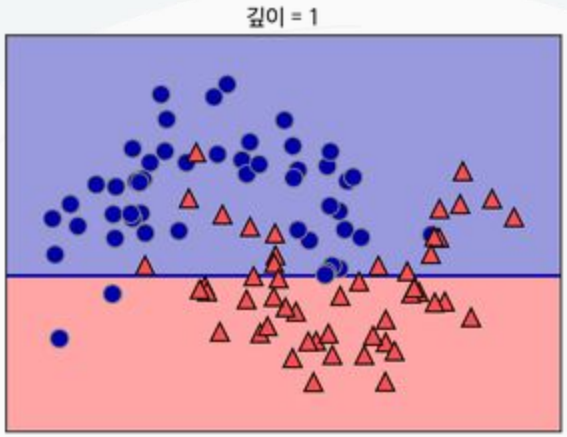
02

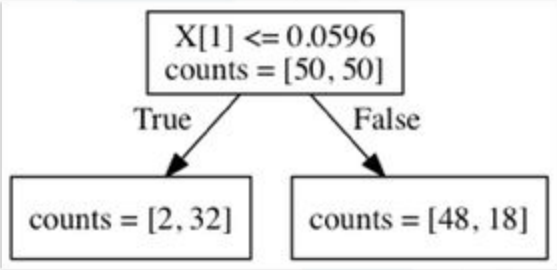
결정 트리 (Decision Tree)

결정 트리(Decision Tree, 의사결정트리, 의사결정나무라고도 함)는 분류(Classification)와 회귀(Regression) 모두 가능한 지도 학습 모델 중 하나입니다. 결정 트리는 스무고개 하듯이 예/아니오 질문을 이어가며 학습합니다. 매, 펭귄, 돌고래, 곰을 구분한다고 생각해봅시다. 매와 펭귄은 날개를 있고, 돌고래와 곰은 날개가 없습니다. '날개가 있나요?'라는 질문을 통해 매, 펭귄 / 돌고래, 곰을 나눌 수 있습니다. 매와 펭귄은 '날 수 있나요?'라는 질문으로 나눌 수 있고, 돌고래와 곰은 '지느러미가 있나요?'라는 질문으로 나눌 수 있습니다.



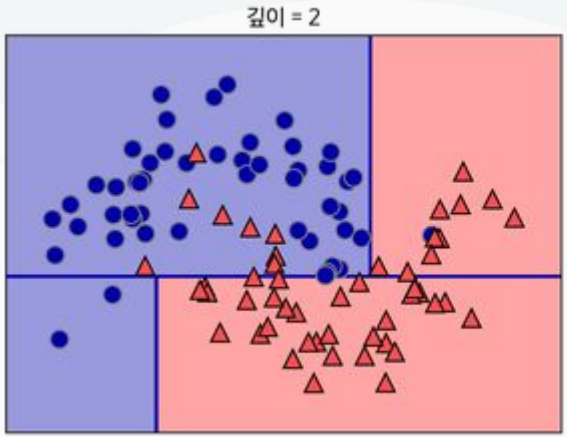
결정 트리 알고리즘의 프로세스를 간단히 알아보겠습니다.

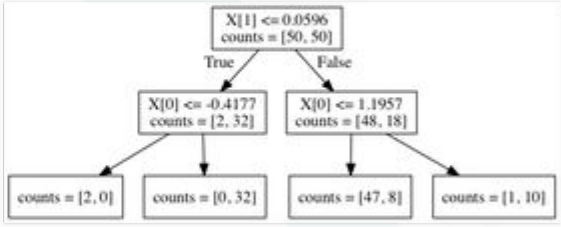




결정 트리 (Decision Tree)

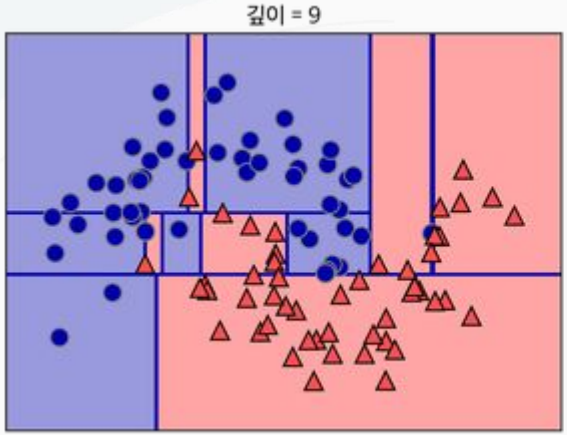
먼저 위와 같이 데이터를 가장 잘 구분할 수 있는 질문을 기준으로 나눕니다.

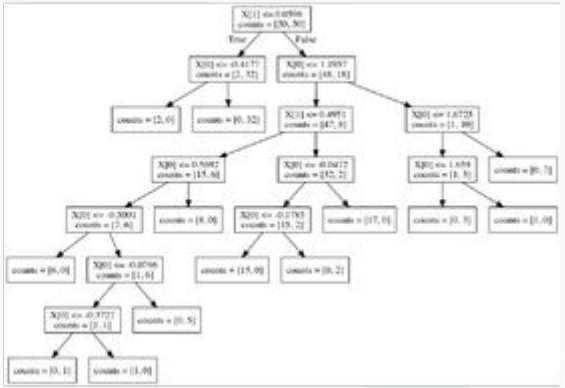




결정 트리 (Decision Tree)

나뉜 각 범주에서 또 다시 데이터를 가장 잘 구분할 수 있는 질문을 기준으로 나눕니다. 이를 지나치게 많이 하면 아래와 같이 오버피팅이 됩니다. 결정 트리에 아무 파라미터를 주지 않고 모델링하면 오버피팅이 됩니다.





02 결정 트리 - 같이하기 (Decision Tree)

KOREATECH 한국기술교육대학교

KOREATECH HRDI 능력개발교육원

같이하기

이 데이터(매, 펭귄, 돌고래, 곰)를 가지고 예제를 만들어서 간단한 결정트리 모델을 학습, 예측 해 보도록 하겠습니다.

가장 처음으로 라이브러리를 임포트 합니다.

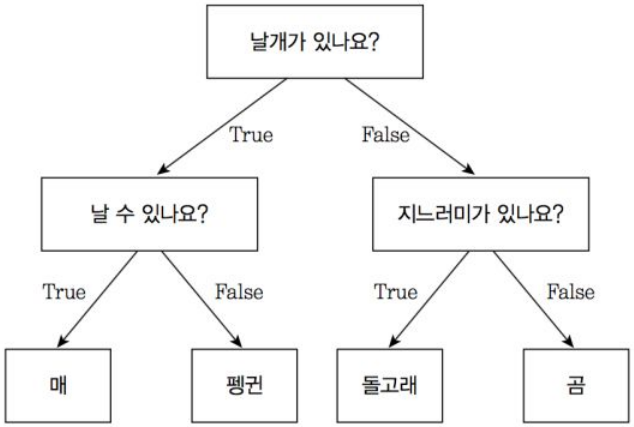
```
[1]  from sklearn.tree import DecisionTreeClassifier
```

같이하기

학습데이터 만들기

- Yes, True -> 1 / No, False -> 0
- N/A(해당사항 없음) -> -1

날개가 있나요?	날 수 있나요?	지느러미가 있나요?	답
1	1	-1	매(1)
1	0	-1	펭귄(2)
0	-1	1	돌고래(3)
0	-1	0	곰(4)



같이하기

학습데이터 만들기

```
# column 1 - 날개가 있나요?
# column 2 - 날 수 있나요?
# column 3 - 지느러미가 있나요?
x_train = [[1,1,-1],[1,0,-1],[0,-1,1],[0,-1,0]]
# 매 - 1, 펭귄 - 2, 돌고래 - 3, 곰 - 4
y_train = [[1],[2],[3],[4]]
```

날개가 있나요?	날 수 있나요?	지느러미가 있나요?	답
1	1	-1	매(1)
1	0	-1	펭귄(2)
0	-1	1	돌고래(3)
0	-1	0	곰(4)

같이하기

학습데이터 만들기

print() 함수를 사용해서 학습 데이터가 잘 만들어 졌는지 확인합니다.

```
[3]  print(x_train)
     print(y_train)
```

```
[[1, 1, -1], [1, 0, -1], [0, -1, 1], [0, -1, 0]]
[[1], [2], [3], [4]]
```

모델 만들기

DecisionTreeClassifier()?

sklearn.tree.DecisionTreeClassifier

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

```
DecisionTreeClfModel = DecisionTreeClassifier()
```

*sklearn에서는 의사결정트리 알고리즘을 구현하는 라이브러리를 제공한다. DecisionTreeClassifier()를 사용하여 의사결정트리 알고리즘을 구현할 수 있다.

모델 만들기

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

아래 링크를 참조해서 각 파라미터를 간단히 설명해 보겠습니다. (+ ~5 pages)

http://june3471.pythonanywhere.com/pages/deeplearning/26/

학습하기

```
DecisionTreeClfModel.fit(x_train,y_train)
```

fit()

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier.fit

학습하기

```
DecisionTreeClfModel.fit(x_train,y_train)
```

X{array-like, sparse matrix} of shape (n_samples, n_features)
The training input samples. Internally, it will be converted to `dtype=np.float32` and if a sparse matrix is provided to a sparse `csc_matrix`.

학습하기

```
DecisionTreeClfModel.fit(x_train,y_train)
```

yarray-like of shape (n_samples,) or (n_samples, n_outputs)
The target values (class labels) as integers or strings.

예측하기

```
#날개가 있고, 날수 없음 => 펭귄 - 2
x_test = [[1, 0, -1]]
```

```
y_pred = DecisionTreeClfModel.predict(x_test)
print(y_pred)
```

예측하기

```
#날개가 있고, 날수 없음 => 펭귄 - 2
x_test = [[1, 0, -1]]
```

날개가 있나요?	날 수 있나요?	지느러미가 있나요?	답
1	1	-1	매(1)
1	0	-1	펭귄(2)
0	-1	1	돌고래(3)
0	-1	0	곰(4)

```
y_pred = DecisionTreeClfModel.predict(x_test)
print(y_pred)
```

```
[2]
```

예측하기 ep. 2

날수 있는지 없는지 정확하게 알 수 없을때.

- 50%만 정확할때
- 80%정도 정확할때

날개가 있나요?	날 수 있나요?	지느러미가 있나요?	답
1	1	-1	매(1)
1	0	-1	펭귄(2)
0	-1	1	돌고래(3)
0	-1	0	곰(4)

예측하기 ep. 2

날 수 있는지 없는지 정확하게 알 수 없을때.

- **50%만 정확할때**

날개가 있나요?	날 수 있나요?	지느러미가 있나요?	답
1	1	-1	매(1)
1	0	-1	펭귄(2)
0	-1	1	돌고래(3)
0	-1	0	곰(4)

```
x_test2 = [[1, 0.5, -1]]
y_pred = DecisionTreeClfModel.predict(x_test2)
print(y_pred)
```

```
[2]
```

예측하기 ep. 2

날 수 있는지 없는지 정확하게 알 수 없을때.

- **80%정도 정확할때**

날개가 있나요?	날 수 있나요?	지느러미가 있나요?	답
1	1	-1	매(1)
1	0	-1	펭귄(2)
0	-1	1	돌고래(3)
0	-1	0	곰(4)

```
x_test2 = [[1, 0.8, -1]]
y_pred = DecisionTreeClfModel.predict(x_test2)
print(y_pred)
```

```
[1]
```

결정 트리의 장점과 단점

장점

- 쉽고 직관적입니다.
- 각 피처의 스케일링과 정규화 같은 전처리 작업의 영향도가 크지 않습니다.

단점

- 규칙을 추가하며, 서브트리(sub-tree)를 만들어 나갈수록 모델이 복잡해지고, 과적합에 빠지기 쉽습니다.

  → 트리의 크기를 사전에 제한하는 튜닝이 필요합니다.

Making a Model In Real World

- 실제로 의사결정트리 그래프를 기반으로 학습 세트를 만드는 것은 매우 드뭅니다.
- 여러가지 모델을 가지고 '**학습 → 예측 → 모델 수정 → 학습자료 수정 → 학습**'을 반복하게 됩니다.

03 결정 트리 - 실습 (Decision Tree)

KOREATECH 한국기술교육대학교

KOREATECH 능력개발교육원

학습 데이터 만들기

직접 학습데이터를 수집해 봅니다.

소개팅에 나간 그 사람이 성공할 수 있을까요?

총합:1.5

예제)

ID	성격	경제력	외모	성공
1	0.5	0.2	0.8	1
2	0.8	0.4	0.3	1
3	0.3	0.4	0.8	0
4	0.1	0.7	0.7	0
5	0.8	0.6	0.1	1
...	...	...	...	...

학습 데이터 만들기

```
# [column 0 - ID 고유번호는 포함시키지 않음]
# column 1 - 성격 # column 2 - 경제력 # column 3 - 외모 # column 4 - 성공

x_train = [
           [0.1, 1  , 0.4], # ID = 1
           [0.2, 0.8, 0.5], # ID = 2
           [0.3, 0.6, 0.6], # ID = 3
           [0.4, 0.4, 0.7], # ID = 4
           [0.5, 0.1, 0.9], # ID = 5
           [0.6, 0.1, 0.8], # ID = 6
           [0.7, 0.3, 0.5], # ID = 7
           [0.8, 0.5, 0.2], # ID = 8
           [0.9, 0.4, 0.1], # ID = 9
           [1  , 0.1, 0.4]  # ID = 10
           ]

y_train = [
           [], # ID = 1
           [], # ID = 2
           [], # ID = 3
           [], # ID = 4
           [], # ID = 5
           [], # ID = 6
           [], # ID = 7
           [], # ID = 8
           [], # ID = 9
           []  # ID = 10
          ]
```

불러오기(import)

1. sklearn.tree 에서 DecisionTreeClassifier를 불러옵니다.
2. Alias(as) 없이 이름은 그대로 사용합니다.

학습 데이터 만들기

직접 학습데이터를 수집해 봅니다.

파일명: *6 머신러닝 이해하기 - 이론과 실습 2.ipynb*

[총합:1.5]

- x_train - feature
- y_train - class / output / answer

학습 데이터 만들기

1. print() 함수를 사용하여 x_train 데이터를 확인
2. print() 함수를 사용하여 y_train 데이터를 확인

- 개수와 정렬이 정확한지 확인이 중요

모델 만들기

1. 변수의 이름 = DecisionTreeClfModel
2. DecisionTreeClassifier() 함수를 사용
3. DecisionTreeClfModel이 잘 만들졌는지 확인
   a. print() 함수 사용

학습하기

1. DecisionTreeClfModel 모델의 fit() 함수를 사용
2. 첫번째 parameter(매개변수)는 feature(속성) 데이터가 들어가야 함으로 x_train
3. 두번째 parameter를 output/label 데이터가 들어가야 함으로 y_train
4. 어떤 에러도 없는 경우는 아래와 같이 출력됩니다.

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

예측하기

1. 변수이름 = x_test
2. 예측하고 싶은 데이터를 생성
   a. 총합계: 1.5
   b. 2차원 list 배열을 사용
3. #2의 데이터를 x_test에 assign

```
# column 1 - 성격
# column 2 - 경제력
# column 3 - 외모
```

4. 변수이름 = y_pred
5. DecisionTreeClfModel의 predict() 함수를 사용해서 나온값을 y_pred에 assign
   a. parameter에 #3에서 만든 x_test를 사용
6. print() 함수를 사용해서 y_pred이 예측된 값을 사용합니다.

7. 1-6를 예측하고 싶은 데이터를 바꿔서 실행

Reference

https://www.fiverr.com/azizpresswala/create-machine-learning-or-deep-learning-models

https://towardsdatascience.com/how-to-build-a-machine-learning-model-439ab8fb3fb1

https://srnghn.medium.com/machine-learning-trying-to-predict-a-numerical-value-8aafb9ad4d36

https://towardsdatascience.com/is-accuracy-everything-96da9afd540d

https://steemit.com/kr/@gillime/r-4-decision-tree

https://ko.wikipedia.org/wiki/%EA%B2%B0%EC%A0%95_%ED%8A%B8%EB%A6%AC_%ED%95%99%EC%8A%B5%EB%B2%95

https://gomguard.tistory.com/86

https://bkshin.tistory.com/entry/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-4-%EA%B2%B0%EC%A0%95-%ED%8A%B8%EB%A6%ACDecision-Tree

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier.fit

https://injo.tistory.com/15