<chapter 15>

**머신러닝 모델 만들기 = 함수 만들기**

: 함수에 값을 입력하면 규칙에 따라 계산한 값을 출력하듯,

머신러닝 모델도 값을 입력하면 정해진 규칙에 따라 계산한 예측값을 출력함.

머신러닝 모델이 함수와 다른점은 만드는 방법임.

함수를 만들 땐, 사람이 계산 규칙을 정해 입력해야 하지만,

머신러닝 모델을 만들 땐, 컴퓨터가 데이터에서 패턴을 찾아 스스로 규칙을 정하게 함

Ex) 환자의 정보를 입력하면 당뇨병 발병 여부를 예측하는 모델을 만든다면,

사람이 해야 할 일) 여러 환자의 정보와 당뇨병 발병 여부 데이터를 수집해 머신러닝 알고리즘에 입력하는 것 뿐.

컴퓨터가 할 일) 환자의 정보와 당뇨병 발병 간의 패턴을 찾아낸 다음, 환자의 정보를 입력하면 당뇨병 발병 여부를 출력하는 모델을 만듦.

어떤 사람의 당뇨병 발병 여부를 알고 싶다면 환자의 정보를 모델에 입력하기만 하면 됨

**머신러닝 모델을 만들 때 필요한 변수 종류**

1. 예측 변수(predictor variable)

: 예측하는데 활용하는 변수 또는 모델에 입력하는 값을 예측하는 변수

1. 타겟 변수(target variable)

: 예측하고자 하는 변수 또는 모델이 출력하는 값

Ex) 환자의 성별, 나이, 흡연 여부, 음주 여부로 당뇨병 발병을 예측하는 모델을 만든다면,

성별, 나이, 흡연 여부, 음주 여부 = 예측 변수 / 당뇨병 발병 여부 = 타겟 변수

**머신러닝 모델을 이용해 미래 예측하기**

: 과거의 값 = 예측 변수 / 미래의 값 = 타겟 변수로 해서 미래를 예측하는 모델을 만들 수 있음.

Ex) 현재 환자의 신체 정보를 입력하면 3년 뒤의 당뇨병 발병 여부를 예측하는 모델을 만들 수 있음.

: 다양한 사업 영역에서 미래를 예측하는데 활용됨.

* 온라인 커머스는 고객이 구매할 가능성이 높은 상품을 예측해서 추천하고,

마케팅사는 서비스에 가입할 가능성이 높은 이용자를 예측해 홍보 전화를 검.

* 금융사는 고객이 대출금을 제때 상환할지 예측해 가입 승인 여부를 결정하고, 반도체 공장은 설비 고장을 예측해 설비가 고장 나기 전에 미리 정비함
* 미래를 예측하면 부가 수익을 창출하고 비용을 줄일 수 있음.

**의사결정나무 모델**

: 순서대로 주어진 질문에 yes/no로 답하면 마지막에 결론을 얻는 구조

Ex) 당뇨병 발병 여부를 예측하는 모델

예측변수 : 흡연여부, 음주여부, 성별, 나이

타겟변수 : 당뇨병 발병 여부

1단계) 타겟변수를 가장 잘 분리하는 예측변수 선택하기

: 의사결정나무 알고리즘은 ‘타겟변수를 가장 잘 분리해 주는 예측변수’를 찾아 첫 번째 질문으로 삼음

**<예측변수가 타겟변수를 분리해주는 정도 파악하는 절차>**

1. 모든 예측 변수를 yes/no로 답할 수 있는 질문으로 만듦.

Ex)흡연을 하십니까? / 음주를 하십니까? / 남자입니까? Or 여자입니까?

1. 모델을 만드는데 사용할 데이터를 앞에서 만든 각각의 질문에 대입한 다음 ‘yes’로 답한 데이터만 추출
2. 추출한 데이터 중 발병인과 정상인의 비율을 구함.

발병인과 정상인의 비율 차이가 크면 클수록,

예측변수가 타겟변수를 잘 분리해낸다고 볼 수 있음.

2단계) 첫 번째 질문의 답변에 따라 데이터를 두 노드로 분할하기

Cf)노드 : 질문의 답변이 같아서 함께 분류된 집단. 의사결정나무 도식에 사각형으로 표현됨.

3단계) 각 노드에서 타겟변수를 가장 잘 분리하는 예측변수 선택하기

: 각 노드에서 타겟변수를 가장 잘 분리해 주는 예측변수를 선택함. 1단계 작업을 노드별로 반복하는 것.

흡연여부는 이미 사용했으니 제외하고 나머지 변수 중에서 타겟 변수를 가장 잘 분리해 주는 두번째 예측변수를 찾으면 됨.

4단계) 노드가 완벽하게 분리될 때까지 반복하기

: 노드에 한 범주만 남아 완벽하게 분리될 때까지 변수를 선택하고 노드를 분할하는 과정을 반복함.

노드에 발병과 정상 중 한 쪽 범주만 남으면 분할을 종료함.

**크로스 밸리데이션(cross validation : 교차검증)**

: 데이터를 분할해 일부는 모델을 만들 때 사용하고, 나머지는 평가할 때 사용하는 방법

**트레이닝 세트(training set : 훈련세트)**

: 분할한 데이터 중에서 모델을 만들 때 사용하는 데이터

**테스트 세트(test set : 시험 세트)**

: 성능을 평가할 때 사용하는 데이터

**컨퓨전 매트릭스(confusion matrix : 혼동행렬)**

: 모델이 예측한 값 중 맞은 경우와 틀린 경우의 빈도를 나타낸 행렬

<성능평가지표 구하기>

**Accuracy(정확도)** : 모델이 예측해서 맞춘 비율

: (TP+TN)/(TP+TN+FP+FN)

: sklearn.metrics의 accuracy\_score( ) 이용

**Precision(정밀도)** : 모델이 관심 클래스를 예측해서 맞춘 비율

: TP/(TP+FP)

: metrics.precision\_score( ) 이용

: 관심 클래스가 분명할 때 사용

**Recall(재현율)** : 모델이 실제 데이터에서 관심 클래스를 찾아낸 비율

: TP/(TP+FN)

: metrics.recall\_score( ) 이용

: 관심 클래스를 최대한 많이 찾아내야 할 때

**F1 score** : recall과 precision의 조화평균

: 0~1 사이의 값을 가지며, 성능이 높을수록 1에 가까운 값이 됨

: accuracy와 달리 타겟 변수의 클래스가 불균형해도 모델의 성능을 잘 표현함

: metrics.f1\_score( )을 이용

: recall과 precision이 모두 중요할 때

<chapter 16>