〈0325 인공지능개론〉

· Data 수집시 고려할 사항

- 결측치 제거
- outlier 제거
- label 간의 Data balance를 잘 맞추기
  - 편향을 없애기 위해서

〈Ch03. 머신러닝의 기초〉

4 붓꽃을 머신러닝으로 분류해보자

· Machine Learning 과정

- 데이터세트 읽어 들이기
- 특징과 레이블
- 훈련데이터와 테스트 데이터
- 모델 선택
  - KNN 알고리즘(K-Nearest Neighbor Algorithm.): 가장 가까운 k개의 이웃에 의존하는 분류 방법
    - $k > 2$, k == odd
    - KNN elbow: K를 결정하는 방법
      - 이웃수가 증가하다 성능향상이 둔해지는 지점
- 학습
- 예측 및 평가
- 적용

5 필기체 숫자 이미지를 분류해보자

- Machine Learning 과정
  - 데이터 세트 읽어들이기
  - 훈련데이터와 테스트데이터
  - 모델
  - 학습
  - 예측 및 평가

6 머신러닝 알고리즘의 성능평가

- 정확도(accuracy) : $\frac{\text{올바르게 분류한 샘플 수}}{\text{전체 샘플 수}}$
  - class가 imbalance 일 경우, 문제 발생

- 혼동행렬(Confusion Matrix) : 학습된 머신러닝 시스템이 예측을 하면서 얼마나 혼동하고 있는지 나타내는 행렬
  - TP : 긍정을 올바르게 예측 (원래 True)
  - FP : 부정을 잘못 예측 (원래 False)
  - TN : 부정을 올바르게 예측 (원래 False)
  - FN : 긍정을 잘못 예측 (원래 True)
  - 민감도 (Sensitivity) : 양성인 샘플 중 양성으로 예측한 샘플의 비율
    - $\frac{TP}{TP+FN}$
  - 특이도(Specificity) : 음성인 샘플 중 음성으로 예측한 샘플의 비율
    - $\frac{TN}{TN+FP}$
  - 정밀도(Precision) : 양성으로 예측된 샘플 중 실제 양성인 비율
    - $\frac{TP}{TP+FP}$
  - F1 Score : 정밀도와 재현율의 조화평균
    - $\frac{precision * sensitivity}{precision + sensitivity}$

7 머신러닝의 용도

- 명시적 알고리즘을 설계하고 프로그래밍하는 것이 어렵거나 불가능한 경우

<Ch.04 선형 회귀>

1 선형 회귀

- 회귀 (regression): 일반적으로 데이터들을 고차원 공간에 찍은 후에 이들 데이터들을 가장 잘 설명하는 직선에서 곡선을 찾는 문제

- 선형회귀 (linear regression): 선형 모델을 사용하여 회귀문제를 푸는 것
  - $f(x) = wx + b$
    - w (weight): 가중치
    - b (bias): 바이어스

- 학습과 손실
  - ML에서 learning이란 훈련데이터로부터 손실을 최소화 하는 weight와 bias를 학습하는 것

2 선형회귀에서 손실함수 최소화 방법

- 경사하강법: 나중에 설명.

3 선형회귀 파이썬 구현 #1

- 점 3개를 주고 경사하강법 구현해보기

4 선형 회기 파이썬 구현 #2

- 사이킷런 라이브러리를 사용하여 회귀함수를 구현

5 과잉적합 vs 과소적합

- 과잉적합(overfitting): 학습하는 데이터에서는 성능이 뛰어나지만 새로운 데이터에서는 성능이 안나옴
  - train data의 noise까지 학습했기 때문에

- 과소적합(underfitting): 훈련데이터에서도 성능이 좋지 않은 경우
  - train data의 양이 너무 적기 때문에
  - 모델이 너무 단순해 train data의 패턴을 충분히 잡아내지 못함

- 4/15 실기시험(Github 참조 가능)