Study Title Day 5

Statistic/Math

1 R square의 의미는 무엇인가요?
2 평균(mean)과 중앙값(median)중에 어떤 케이스에서 뭐를 써야할까요?

• Machine Learning

3 회귀 / 분류시 알맞은 metric은 무엇일까?
4 Association Rule의 Support, Confidence, Lift에 대해 설명해주세요.
5 최적화 기법중 Newton's Method와 Gradient Descent 방법에 대해 알고 있나요?

• Deep Learning

6 Gradient Descent에 대해서 쉽게 설명한다면?
7 왜 꼭 Gradient를 써야 할까? 그 그래프에서 가로축과 세로축 각각은 무엇인가? 실제 상황에서는 그 그래프가 어떻게 그려질까?
8 GD 중에 때때로 Loss가 증가하는 이유는?
9 Back Propagation에 대해서 쉽게 설명 한다면?

• Python

9 What is self in Python?

• Algorithm

10 다음 코드의 출력 값은?

```
void func(int i, string s){
  if(s.length() == 0) return;
  if(i % 2 == 0){
     printf("%c", s[s.length()-1]);
     func(i+1, s.substr(0, s.length()-1));
  } else{
     printf("%c", s[0]);
     func(i+1, s.substr(1));
  }
  return;
}

int main(void){
  func(0,"abcd");
}
```

'dacb'

1. R square(결정계수): 회귀 모델에서 독립변수가 종속 변수를 얼마나 설명하는지 가리키는 지표 (설명력)
   ↳ 0~1 사이이며, 1이면 회귀선으로 모든 데이터 표현 가능

2. 평균은 분산이 작은 데이터에 적합, 중앙값은 극단적인 값에 크게 영향을 받지 않는다.

3. metric (척도) : 학습을 통해 목표를 얼마나 잘(못) 달성했는지를 나타내는 지표
   - 회귀문제 metric : 실제 값과 모델이 예측하는 값의 차이에 기반을 둔 metric 사용 : MSE, MAE, R-Square
   - 분류 문제 metric : 어떤 모델이 얼마나 데이터 클래스에 맞게 분류했느냐를 측정하기 위해 혼돈 행렬 사용 (정확도, 정밀도, 재현율)

		예측 클래스 (predict class) False	예측 클래스 (predict class) True
실제 클래스 (Actual Class)	False	TN (True Negative)	FP (False Positive)
	True	FN (False Negative)	TP (True Positive)

: 정답
: 오답

- 정확도(accuracy): 모델이 얼마나 데이터를 잘 분류했느냐 ($\frac{(TP+TN)}{(TP+TN+FP+FN)}$)
- 정밀도(precision): 모델이 True라고 분류한 것 중 실제로 True인 데이터 비율 ($\frac{TP}{(TP+FP)}$)
- 재현율(Recall): 실제 True인 것 중 모델이 True로 분류한 비율 ($\frac{TP}{(FN+TP)}$)

Study Title Day 5

4. Association Rule(연관 규칙): 어떤 사건이 얼마나 자주 함께 발생하는지, 서로 얼마나 연관되어 있는지 표시하는 것

Support: 전체 경우의 수에서 두 아이템이 같이 나오는 비율 $\text{Support}(X \Rightarrow Y) = \frac{N_{XUY}}{N}$

Confidence: X가 나온 경우 중 X와 Y가 함께 나올 비율 $\text{Confidence}(X \Rightarrow Y) = \frac{N_{XUY}}{N_X}$

Lift: X와 Y가 같이 나오는 비율을 X가 나올 비율과 Y가 나올 비율의 곱으로 나눈 값 $\text{Lift}(X \Rightarrow Y) = \frac{N_{XUY}/N}{(N_X/N)\cdot(N_Y/N)} = \frac{N_{XUY}\cdot N}{N_X \cdot N_Y}$

5. Gradient Descent: 초기값부터 경사를 따라 천천히 내려가서 최적점($f'=0$)을 찾아나가는 방식

Newton's Method: $f(x)=0$의 해를 근사적으로 찾는 방법 $x^{t+1} = x^t - \frac{f(x^t)}{f'(x^t)}$, 극소를 찾기 위해서 f'' 이용

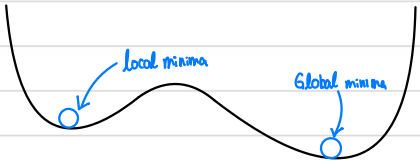
6. 어떤 함수의 Local Minima를 찾기위해 Gradient 반대방향으로 이동하는 방법

Loss function을 미분하여 그 기울기값을 구하고, 경사가 하강하는 방향으로 파라미터 값을 점진적으로 이동하여 탐색

7. 차원이 복잡해지면 GD같이 직관적인 방법이 계산량 측면에서 효율적임

x축: 모델의 파라미터 θ, y축: 손실함수의 결과값

8. local minima에 접근한 경우, Loss 증가



9. Back Propagation

· Loss에 대한 입력값의 기울기(미분값)를 출력층 layer에서부터 계산하여 거꾸로 전파시키는 것

거꾸로 전파시켜서 최종적으로 출력층에서의 output값에 대한 입력층에서의 input data의 기울기를 구할 수 있다.

10. Python의 self: 인스턴스 메서드의 첫번째 인자로, 호출될 때 self에 인스턴스를 넣고 참조하여 실행한다.