제2장 지도학습 (Supervised Learning)

앞에서 살펴보았듯이 지도학습 머신 러닝은 가장 널리 쓰이는 성공적인 타입의 머신 러닝이다. 이번 장에서 우리는 supervised learning 에 대해 좀 더 깊이 다루고, 몇 가지 유명한 알고리즘들에 대해 살펴볼 것이다. 우리는 이미 1장에서 붓꽃의 물리적인 측정치를 이용하여 그 종을 분류하는 지도학습 머신 러닝 애플리케이션을 경험했다.

지도학습은 주어진 입력 데이터로부터 사전에 정해진 어떤 특정한 결과를 예측하길 원하고, 입력/출력 데이터 쌍을 가지고 있을 때 사용된다는 것을 기억할 것이다. 우리는 트레이닝셋으로 불리는 입력/출력 데이터로부터 머신 러닝 모델을 만들었다. 우리의 목적은 이전에 보지 못한 새로운 데이터에 대해서 정확한 예측값을 도출하는 것이다.

분류 분석과 회귀 분석

지도학습 머신 러닝에서 주로 사용되는 두 가지 타입은 분류분석(Classification)과 회귀분석(Regression)이다.

분류분석의 목적는 클래스 레이블을 예측하는 것이다. 클래스 레이블은 도출되기를 원하는 결과를 사전에 정의해 놓은 목록이다. 1장에서 우리는 선택가능한 세 가지 종 중의 하나로 붓꽃의 종을 분류했다. 분류분석은 종종 이진 분류분석(binary classification)과 다중클래스 분류분석(multiclass classification)으로 나뉜다. 이진 분류는 단 두 개의 클래스 중 하나로 구분하는 특수한 경우이며, 다중클래스 분류의 클래스는 두 개 이상이다. 이진 분류분석은 단순히 질문에 “예/아니오”로 답하는 것이라 생각하면 된다. 이메일이 스팸인지 아닌지 분류하는 것을 그 예로 들 수 있다. 반면에, 붓꽃의 종을 분류하는 예는 다중클래스 분류분석이라고 할 수 있다. 특정 웹사이트에 사용된 언어를 분류하는 일을 또 다른 예로 들 수 있다. 선택 가능한 대상 언어들은 물론 미리 정의되어 있을 것이다.

회귀분석의 목적은 연속된 숫자, 또는 수학용어로 실수값을 예측하는 것이다. 어떤 사람의 교육 수준, 나이, 주거지에 따라 그의 연수입을 예측하는 것은 회귀분석의 한 예이다. 예측된 값은 주어진 범위내의 어떠한 값이라도 될 수 있다. 회귀분석의 또 다른 예로는 옥수수 농장의 지난해 생산량, 날씨, 일꾼의 수를 근거로 올해의 생산량을 예측하는 것이다. 생산량은 임의의 어떤 수라도 될 수 있다.

회귀분석과 분류분석을 구별하는 가장 쉬운 방법은 결과값에 연속성이 있는지 없는지를 보는 것이다. 만약 도출 가능한 여러 결과값들 사이에 연속성이 있으면 회귀분석 문제라고 할 수 있다. 어떤 사람의 연수입 금액에는 확실한 연속성이 있다. 그의 1년 수입이 40,000 달러이든, 40,001 달러이든 실제적인 차이는 미미하다. 즉 40,000 달러로 예측이 나와야하는 알고리즘에서 결과가 39,999 또는 40,001 어느 쪽으로 나와도 큰 문제가 되지는 않는다.

하지만, 웹사이트에 사용된 언어를 판별하는 문제에 있어서는(이것은 분류분석 문제이다) 매우 다르다. 한 웹사이트는 A 또는 B 언어로 되어 있다. 두 언어 사이에는 연속성이라는 개념이 있을 수 없다. 이를테면 프랑스어와 영어 사이에 해당하는 언어란 존재하지 않는다.

Generalization, Overfitting, and Underfitting

지도학습에서 우리는 트레이닝 데이터를 기반으로 모델을 만들고, 그 다음에 우리가 사용한 트레이닝셋과 동일한 특성을 가진 새로운 데이터에 대한 예측을 한다. 만약 그 모델이 새로운 데이터에 대해 정확한 예측을 할 수 있다면, 우리는 트레이닝셋으로부터 테스트셋으로 일반화(generalize)가 가능하다고 말할 수 있다. 우리가 원하는 것은 가능한 한 정확하게 일반화된 모델을 만들어 내는 것이다.

일반적으로 우리는 트레이닝 데이터에 맞추는 방식으로 모델을 만든다. 만약 트레이닝셋과 테스트셋이 서로 공통적인 요인을 충분히 가지고 있다면 모델의 테스트셋에 대한 예측이 정확할 것이라고 기대할 수 있다. 하지만, 그렇지 못한 경우가 종종 있다.

이 점에 대한 자세한 설명을 위해 만든 예제를 살펴보자. 신참 데이터 사이언티스트가 이전에 보트를 구매한 고객과 그렇지 않은 고객에 대한 데이터를 바탕으로 어느 고객이 보트를 구매할지 예측하려고 한다. 실제 구매가 이루어질 가능성이 높은 사람들에게 프로모션 이메일을 보내야 하지만, 전혀 관심이 없는 사람들에게 까지 보내서 귀찮게 하면 안 된다.

아래의 고객 테이블을 보자.

<테이블 이미지>

테이블2-1. 보트 구매 관련 고객 데이터

잠시 데이터를 살펴본 신참 데이터 사이언티스트는 다음과 같은 규칙을 생각해냈다: "45세 이상, 자녀 3명 미만이며 이혼하지 않은 상태인 고객은 보트를 구매하고 싶어한다." 이 규칙이 얼마나 정확한지 질문을 한다면 이 데이터 사이언티스트는 "100% 정확합니다"라 대답할 것이다. 사실, 테이블에 나타난 데이터를 보면 이 규칙은 정확하다. 데이터상 같은 나이가 두번 나타나는 경우는 없기 때문에 나이가 66, 52, 53, 58세인 고객들은 보트를 구매하길 원할 것이고 나머지는 아니라고 말할 수 있다.

이 데이터셋을 바탕으로 그 외에도 많은 규칙들을 만들어낼 수 있지만, 이 데이터셋에 대해서 예측을 수행하는 것이 목표가 아님을 기억해야 한다. 이 고객들에 대한 답을 우리는 이미 알고 있기 때문이다. 우리가 원하는 것은 신규 고객들에게 보트 구매 의사가 있는지 알아내는 것이다. 우리에게 필요한 것은 새로운 고객들에 대해 잘 동작하는 규칙을 찾는 것이기 때문에, 트레이닝셋에 대해 100%의 정확성을 갖는 모델은 전혀 도움이 되지 않는다. 그 데이터 사이언티스트의 규칙이 새로운 고객에게도 잘 동작하리라고 기대하긴 어려울 것이다.

알고리즘이 새로운 데이터에 대해서도 잘 동작하는지 아닌지 측정하는 유일한 방법은 테스트셋으로 검증하는 것이다. 하지만 우리는 직감적으로 단순한 모델일수록 새 데이터에 대해 좀 더 일반화가 잘 된다는 것을 알 수 있다. “50세 이상인 사람은 보트를 사고 싶어한다”는 규칙은 여기에 자녀 수와 결혼 상태를 추가한 규칙보다 다른 고객들에 대해 더 잘 설명할 것이다. 그렇기 때문에 우리는 항상 가장 단순한 모델을 찾길 원한다. 그래서 “가지고 있는 정보의 양에 비해 모델이 너무 복잡한” 경우 과적합(overfitting) 되었다고 말한다.

모델이 트레이닝셋의 특성에 너무 최적화되어 새로운 데이터에 대해서 일반화를 잘 하지 못할 때 overfitting이 발생한다. 그와 반대로, 만약 당신의 모델이 너무나 단순하다면 (예를 들어 “집을 소유한 모든 사람은 보트 구매를 원한다”) 그 모델은 데이터의 다양한 측면을 제대로 잡아내지 못한 것이며, 트레이닝셋에 대해서도 제대로 동작하지 못할 것이다. 너무 단순한 모델을 선택하는 것을 underfitting 이라고 한다.

우리의 모델을 조금 더 복잡하게 만들면 트레이닝 데이터에 대한 예측력은 좋아질 것이다. 하지만 이 모델이 지나치게 복잡해진다면 트레이닝셋의 각 개별 데이터 포인트에 너무 집중하게 되어서 새로운 데이터에 대해서는 일반화가 전혀 되지 않을 것이다.

여기서 최상의 일반화 성능(generalization performance)을 내는 sweet spot이 있음을 알 수 있다. 이 지점이 우리가 찾는 모델이다. 그림 2-1은 overfitting과 underfitting 사이의 상호 모순적 관계를 나타낸 것이다.

<그림2-1. 모델 복잡도와 트레이닝셋/테스트셋 예측 정확도 간의 관계>

데이터셋의 크기와 모델 복잡도의 관계

모델의 복잡도는 트레이닝 데이터셋에 포함된 입력값의 분포와 매우 밀접한 관계를 가지고 있다. 매우 다양한 데이터 포인트들을 가지고 있다면 과적합을 피하면서 좀더 복잡한 모델을 사용할 수 있다. 일반적으로 데이터의 양이 늘어날수록 데이터의 다양성도 커지기 때문에 대용량의 데이터셋이 좀더 정교한 모델을 만들기에 좋다. 하지만 동일한 데이터 포인트의 중복이 많거나 서로 매우 비슷한 데이터가 많은 것은 도움이 되지 않는다.

앞의 예제로 돌아가서, 만약 우리가 10,000개 이상의 고객 데이터를 가지고 있다면 “45세 이상이고, 자녀가 3명 이하, 이혼하지 않은 상태의 고객이라면 보트 구매에 관심을 가지고 있다.”라는 규칙은 12개 데이터를 사용했을 때보다 훨씬 더 신뢰성이 가질 것이다.

이 책에서 우리는 고정된 크기의 데이터셋을 사용할 것이다. 하지만 실제 업무에서는 얼마나 많은 데이터를 수집해야 하는지 결정하는 능력이 당신의 모델을 개선하는데 많은 도움이 될 것이다. 대량의 데이터가 주는 이점을 절대 과소평가하지 말라.

지도학습 머신 러닝 알고리즘

우리는 이제 가장 많이 알려진 머신 러닝 알고리즘을 살펴보고, 데이터로부터 학습하는 방법과 예측을 수행하는 방법에 대해 설명할 것이다. 또한 모델의 복잡도에 따라 모델이 어떻게 달라지는지 논의하고, 각 알고리즘이 모델 생성 원리에 대해 간략하게 살펴볼 것이다. 각 알고리즘의 장점과 단점, 그리고 어떤 형태의 데이터를 적용하는 것이 좋을지도 알아볼 것이다. 그리고 알고리즘에서 중요한 파라미터들과 옵션들에 대해서도 설명할 것이다.

각 알고리즘에 대한 설명을 너무 상세하게 읽을 필요는 없지만, 그 모델들을 이해하는 것은 머신 러닝 알고리즘들이 서로 다른 방식으로 동작함을 알게 해줄 것이다.

샘플 데이터셋

우리는 알고리즘들을 설명하기 위해 몇가지 데이터셋을 사용할 것이다. 어떤 데이터셋은 알고리즘의 특정한 측면을 강조하기 위해 임의적으로 만든 것이며, 또 다른 것은 사이즈가 큰 실무용 데이터이다.

한 예로 이진 분류를 설명하기 위해 만든 forge 데이터셋은 2개의 피쳐를 가지고 있다. 다음의 코드는 이 데이터셋의 모든 데이터 포인트를 보여주는 산점도(그림2-2)를 만드는 것이다. 산점도의 x축은 첫번째 피쳐이고, y축은 두번째 피쳐이다. 산점도에서는 항상 하나의 데이터 포인트는 하나의 점으로 표현된다. 점의 색깔과 모양은 해당 데이터의 클래스를 가리킨다. X.shape 로 볼 수 있듯이 이 데이터셋은 26개의 데이터 포인트를 가지고 있다.

<코드>

회귀분석 알고리즘을 설명하기 위해서는 wave 데이터셋을 사용할 것이다. wave 데이터셋은 1개의 입력 변수와 1개의 연속형 목표 변수를 가지고 있다. (그림2-3)

<코드>

이렇게 단순하고 저차원의 데이터셋을 사용하는 이유는 그것을 2차원 공간에 시각화하기가 좋기 때문이다. 2개 이상의 피쳐를 가진 데이터는 모든 특성을 다 보여주기 쉽지 않다.

우리는 또한 scikit-learn에 포함되어 있는 실무적인 데이터셋도 함께 사용한다. 그 중 하나는 위스콘신 유방암 데이터셋(앞으로 cancer로 지칭)으로 유방암과 관련된 종양의 의학적인 측정치들이 포함되어 있다. 각 종양은 “benign”(온화한, 무해한 종양) 또는 “malignant”(악성의, 암이 되는 종양)으로 레이블되어 있다.