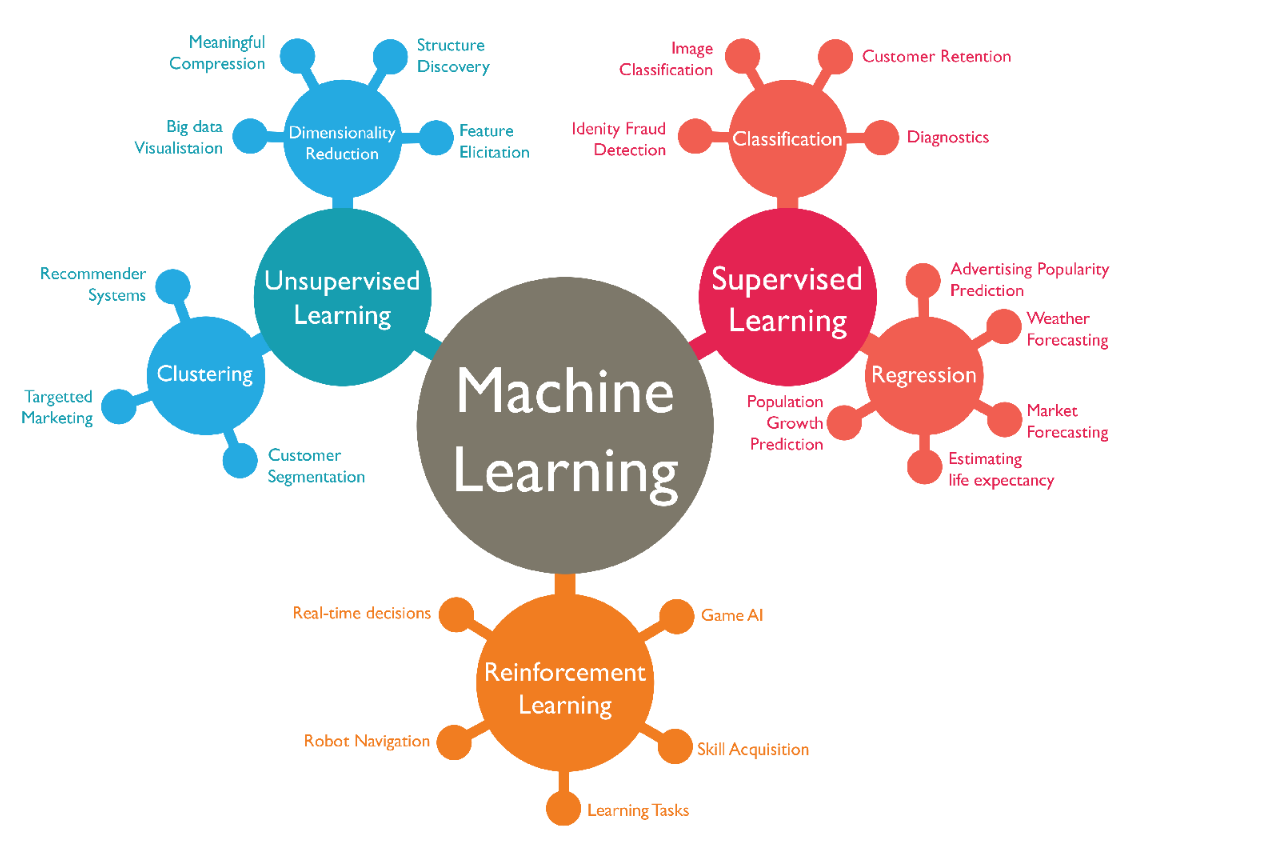
Chapter one: Computers learn from datasets

Variation of learning-supervised learning, unsupervised learning, reinforcement learning



Row: Samples (instance or observation)

Column: features (options, observed numeric data)

Class label: targets (Labels to be classified, regressed, clustered…)

Feature engineering (O’reilly, python data science handbook, feature engineering)

일반적인 머신 러닝 데이터는 깔끔한 Feature-Label형태의 수치적(numerical as N) 데이터만 사용한다. 하지만 범주적(categorical as C) 데이터에는 많은 수치적 데이터에 맞는 모델들이 잘 맞지 않을 것이다.

예시: 도시 신문에 나온 방에 대한 데이터이다. Price(N), room(N), neighborhood(C) 라는 레이블로 이루어져 있다(p. 429). 과연 이웃 이름에 관한 데이터를 수치적으로 표현하고 모델을 만들 수 있을까? 불가능하다. 이웃간 데이터를 수치화하면 이웃-이웃은 하나의 수치적 데이터가 되어 버린다. (예: 홍길동->4, 이준혁->3, 고길동->1, 홍길동-이준혁=고길동??)

이런 경우 해당 범주(이 경우, 이웃)의 유무를 각각 1 혹은 0으로 나타내어 새로운 열을 만들어 주는 one-hot encoding이라는 기법을 사용한다.

또, 신문 기사 텍스트 토막에 어떤 문자가 자주 쓰였는지를 이용해 모델을 제작하는 Wordcount라는 기법도 있는데, 이는 너무 자주 나오는 단어에 너무 많은 가중치를 줄 수 있다. 이 때 단어가 문서에서 얼마나 자주 등장하느냐에 따라 가중치를 결정하는 TF-IDF라는 기법을 사용할 수 있다.

이 외에도 수학적 유도 기법(ex: 스케일 조정, 표준화), 이미지 기법, 누락 데이터 대체 등 많은 기법이 있는데, 데이터셋의 특성을 이해하고 그에 따라 데이터를 바꾸어 가중치가 오버피팅되는 문제에서 벗어나야 한다.

Chapter two: Perceptron and simple classification algorithm

인공 뉴런의 정의

-WXi + b, 특정 샘플 xi의 입력과 임계값 theta의 대소 비교를 통해 특정 클래스나 값을 반환하는 알고리즘.

모든 알고리즘에서 반환값은 결정함수(relu, sigmoid, unit step function)등에 의해 결정된다. 초기 인공 뉴런에서는 비선형 결정 함수로서 계단 함수를 사용했다. 이때 음수 임계값 -θ를 절편(b, bias)이라 한다.

여기서 뉴런은 이전 예측의 결과값 y^에 따라 W값을 업데이트하는데, 그 식은 아래와 같다.

W=W+dW, dW=ε(y-y^)\*X

여기서, 엡실론은 학습률이며, w의 업데이트는 예측 결과값이랑 실제 결과값의 차이와 학습률을 이용해 가중치를 업데이트 함을 알 수 있다. 이후 아달린은 연속 함수로 cost(or loss)를 정의하는 오차를 정의하였으며, 활성화 함수와 결정 함수를 동시에 사용하여 최종 예측을 수행하였다.

아달린은 오차를 제곱 오차합(Sum of squared errors, 현대에는 RMSE, MAE, MSE등의 다양한 오차 결정 식이 사용된다)로 정의하였다. 해당 오차식의 미분(경사)를 최소화하는 방식을 이용해 최적의 가중치를 찾을 수 있었는데 이를 경사 하강법이라 칭한다. (하강의 방법에는 배치 하강법과 확률적 하강법이 존재하나 현대에는 미니 배치 하강법이 주로 이용되고 있다.)