Chapter 1 한눈에 보는 머신러닝

*머신러닝은 명시적인 규칙을 코딩하지 않고 기계가 데이터로부터 학습하여 어떤 작업을 더 잘하도록 만드는 것

*여러종류의 머신러닝 시스템이 있습니다. 지도학습과 비지도 학습, 배치학습과 온라인 학습, 사례 기반학습과 모델 기반 학습 등입니다.

*머신러닝 프로젝트에서는 훈련세트에 데이터를 모아 학습 알고리즘에 주입합니다. 학습 알고리즘이 모델기반이면 훈련세트에 모델을 맞추기 위해 모델 파라미터를 조정하고(즉, 훈련세트에서 좋은 예측을 만들기 위해), 새로운 데이터에도 좋은 예측을 만들거라고 기대합니다. 알고리즘이 사례 기반이면 샘플을 기억하는 것이 학습이고 유사도 측정을 사용하여 학습한 새로운 샘플을 비교하는 식으로 새로운 샘플에 일반화합니다.

*훈련세트가 너무 작거나, 대표성이 없는 데이터이거나, 잡음이 많고 관련 없는 특성으로 오염되어 있다면 시스템이 잘 작동하지 않습니다. 마지막으로 모델이 너무 단순하거나(과소적합) 너무 복잡하지 않아야 합니다.(과대적합)

*머신러닝 정의하기
=>머신러닝은 데이터로부터 학습 할 수 있는 시스템을 만드는 것이다. 학습이란 어떤 작업에서 주어진 성능 지표가 더 나아지는 것을 의미한다.

*머신러닝이 도움을 줄 수 있는 문제 유형
=>명확한 해결책이 없는 복잡한문제, 수작업으로 만든 긴 규칙 리스트를 제공하는 경우, 변화하는 환경에 적응하는 시스템을 만드는 경우, 사람에게 통찰을 제공해야하는경우(데이터마이닝)

*레이블된 훈련세트
=>각 샘플에 대해 원하는 정답(레이블)을 담고 있는 훈련세트

*보편적인 지도학습 작업 네가지
=>군집, 시각화, 차원축소, 연관규칙

*사전정보가 없는 여러 지형에서 로봇을 걸어가게 하려면 어떤 종류의 머신러닝 알고리즘을 사용할까
=>알려지지 않은 지형을 탐험하는 로봇을 학습시키는 가장 좋은 방법은 강화학습이다. 이는 전형적으로 강화 학습이 다루는 유형의 문제이다.

*고객을 여러 그룹으로 분할하려면 어떤 알고리즘을 사용할까
=>만약 그룹을 어떻게 정의할지 모른다면 비슷한 고객끼리 군집으로 나누기 위해 군집 알고리즘(비지도학습)을 사용할 수 있다. 그러나 어떤 그룹이 있어야 할지 안다면 분류 알고리즘(지도학습)에 각 그룹에 대한 샘플을 주입한다. 그러면 알고리즘이 전체 고객을 이런 그룹으로 분류하게 될것이다.

*스팸 감지의 문제는 지도학습과 비지도 학습 중 어떤 문제로 볼수있을까
=>스팸 감지는 전형적인 지도 학습문제이다.

*온라인 학습 시스템이 무엇인가
=>온라인 학습 시스템은 배치 학습 시스템과 달리 점진적으로 학습할 수 있다. 이 방식은 변화하는 데이터와 자율 시스템이 빠르게 적응하고 매우 많은 양의 데이터를 훈련 시킬 수 있다.

*외부 메모리 학습은 무엇인가
=>외부 메모리 알고리즘은 컴퓨터의 주 메모리에 들어갈 수 없는 대용량의 데이터를 다룰 수 있다. 외부 메모리 학습 알고리즘은 데이터를 미니배치로 나누고 온라인 학습 기법을 사용해 학습한다.

*예측을 하기 위해 유사도 측정에 의존하는 학습 알고리즘은 무엇인가
=>사례 기반 학습 시스템은 훈련 데이터를 기억하는 학습입니다. 새로운 샘플이 주어지면 유사도 측정을 사용해 학습된 샘플 중에서 가장 비슷한 것을 찾아 예측으로 사용한다.

*모델 파라미터와 알고리즘의 하이퍼파라미터 사이에는 어떤 차이가 있나
=>모델은하나 이상의 파라미터(예를 들면 선형 모델의 기울기)를 사용해 새로운 샘플이 주어지면 무엇을 예측할지 결정한다. 학습 알고리즘은 모델이 새로운 샘플에 잘 일반화 되도록 이런 파라미터들의 최적값을 찾는다. 하이퍼파라미터는 모델이 아니라 이런 학습 알고리즘 자체의 파라미터이다.(예를 들면 적용할 규제의 정도)

*모델기반 알고리즘이 찾는 것은 무엇인가, 성공을 위해 이 알고리즘을 사용하는 가장 일반적인 전략은, 예측은 어떻게
=>모델 기반 학습 알고리즘은 새로운 샘플에 잘 일반화 되기 위한 모델 파라미터이 최적값을 찾는다. 일반적으로 훈련 데이터에서 시스템의 예측이 아니라 나쁜지 측정하고 모델에 규제가 없다면 모델 복잡도에 대한 페널티를 더한 비용함수를 최소화함으로써 시스템을 훈련시킨다. 예측을 만들려면 학습 알고리즘이 찾은 파라미터를 사용하는 모델의 예측 함수에 새로운 샘플의 특성을 주입한다.

*머신러닝의 주요 도전과제는 무엇인가
=>머신러닝의 주요 도전과제는 부조한 데이터, 낮은 데이터 품질, 대표성 없는 데이터, 무의미한 특성, 훈련 데이터에 과소 적합된 과도하게 간단한 모델, 훈련 데이터에 과대적합된 과도하게 복잡한 모델등 이다.

*모델이 훈련 데이터에서의 성능은 좋지만 새로운 샘플에서 일반화 성능이 나쁘다면 어떤 문제가 있는건가, 해결 해결책 어떤 것이 있는가
=>모델이 훈련 데이터에서는 잘 작동하지만 새로운 샘플에서는 형편없다면 이 모델은 훈련 데이터에 과대 적합되었을 가능성이 높다.(또는 매우 운이 좋은 경우) 과대 적합에 대한 해결책은 더 많은 데이터를 모으거나, 모델을 단순화 하거나, 훈련데이터에있는 잡음을 감소 시키는 것이다.

*테스트 세트가 무엇이고 왜 사용하는가
=>테스트 세트는 실전에 배치되기 전에 모델이 새로운 샘플에 대해 만들 일반화 오차를 추정하기 위해 사용한다.

*검증 세트의 목적은 무엇인가
=>검증 세트는 모델을 비교하는데 사용된다. 이를 사용해 가장 좋은 모델을 고르고 하이퍼파라미터를 튜닝한다.

*훈련-개발 세트가 무엇인가, 언제 필요하고 어떻게 사용해야 하나
=>훈련-개발 세트는 검증, 테스트 세트에 사용되는 데이터와 훈련 세트 사이에 데이터 불일치 위험이 있을 때 사용한다. 훈련세트의 일부에서 모델을 훈련하고 훈련-개발 세트에서 나쁜 성능을 낸다면 아마도 훈련 세트에 과대 적합되었을 가능성이 높다. 훈련 세트와 훈련-개발 세트 양쪽에서 모두 잘 동작하지만 검증 세트에서 성능이 나쁘다면 훈련 데이터와 검증+테스트 데이터 사이에 데이터 불일치가 있을 가능이 높다. 검증+테스트 데이터에 더 가깝게 되도록 훈련 데이터를 개선해야 한다.

*테스트 세트를 사용해 하이퍼파라미터를 튜닝하면 어떤 문제가 생기는가
=>테스트 세트를 사용해 하이퍼파라미터를 튜닝하면 테스트 세트에 과대적합될 위험이 있고 일반화 오차를 낙관적으로 측정하게 된다.(기대보다 나쁜 성능을 낼것이다.)

Chapter 2 머신러닝 프로젝트 처음부터 끝까지

*머신러닝 주요 단계
1. 큰그림을 본다.
2. 데이터를 구한다.
3. 데이터로부터 통찰을 얻기 위해 탐색하고 시각화 한다.
4. 머신러닝 알고리즘을 위해 데이터를 준비한다.
5. 모델을 선택하고 훈련시킨다.
6. 모델을 상세하게 조정한다.
7. 솔루션을 제시한다.
8. 시스템을 론칭하고 모니터링하고 유지 보수한다.

Chapter 3 분류

*성능 측정
=>교차 검증 : 정확도(Accuracy) 측정
=>오차 행렬(confusion matrix) : 정밀도(precision), 재현율(recall) == 민감도(sensitivity) 등으로 행렬 만듬
=>F1점수 : 정밀도와 재현율을 조합하면 만든 점수
=>ROC 곡선 : ROC(receiber operating characteristic, 수신기 조작 특성) 거짓 양성비율에 대한 진짜 양성비율

Chapter 4 모델 훈련

*경사 하강법(gradient descent, GD)
=>여러 종류의 문제에서 최적의 해법을 찾을 수 있는 일반적인 최적화 알고리즘, 기본 아이디어는 비용함수를 최소화 하기 위해 반복해서 파라미터를 조정해 가는 것

*학습률(learning rate)
=>경사하강법에서 중요한 파라미터는 스텝의 크기, 학습률 하이퍼파라미터로 결정된다. 학습률이 너무 낮으면 최적점에 도달은 하겠지만 오래 걸림

*배치 경사 하강법(batch gradient descent)
=>매스텝에서 훈련 데이터 전체를 사용, 매우 큰 훈련세트에서는 아주 느림

*경사하강법은 특성 수에는 민감하지 않음

*수십만개의 특성에 선형 회귀를 훈련시키려면 정규방정식이나 SVD분해보다 경사 하강법을 사용하는 편이 좋음

*적절한 학습률을 찾는 방법 => 그리드 탐색 : 반복 횟수를 제한 해야함

*반복횟수 제한 방법
=>반복 횟수를 아주 크게 지정하고 그레이디언트 벡터가 아주 작아지면, 즉 벡터의 노름이 어떤 허용오차보다 작아지면 경사 하강법이 거의 최솟값에 도달한 것이므로 알고리즘을 중지

*확률적 경사 하강법
=>배치 경사 하강법의 단점인 매 스텝에서 전체 훈련세트를 사용해 그레디언트를 계산 한다는 사실을 보완 할 수 있는 방법
=>매 스텝에서 한 개의 샘플을 무작위로 선택하고 그 하나의 샘플에 대한 그레디언트를 계산 : 속도 빨라짐, 매우 큰 훈련세트에도 적용 가능
=>단점 : 불안정, 알고리즘이 멈추면 좋은 파라미터이지만 최적치는 아님

*미니 매치 경사 하강법(mini-batch gradient descent)
=>미니 배치라고 부르는 임의의 작은 샘플 세트에 대해 그레디언트를 계산
=>장점 : 행렬 연산에 최적화된 하드웨어, 특히 GPU를 사용해서 얻는 성능 향상

*선형 회귀를 사용한 알고리즘 비교

알고리즘	m이 클 때	외부 메모리 학습 지원	n이 클 때	하이퍼 파라미터수	스케일 조정 필요	사이킷런
정규방정식	빠름	No	느림	0	No	N/A
SVD	빠름	No	느림	0	No	LinearRegression
배치 경사 하강법	느림	No	빠름	2	Yes	SGDRegressor
확률적 경사 하강법	빠름	Yes	빠름	>= 2	Yes	SGDRegressor
미니배치 경사 하강법	빠름	Yes	빠름	>= 2	Yes	SGDRegressor

*다항 회귀(polynomial regression)
=>비선형 데이터를 학습하는데 선형 모델을 사용하는 방법

*릿지(ridge) 회귀 == 티호노프 규제
=>규제가 추가된 선형 회귀 버전
=>학습알고리즘을 데이터에 맞추는 것 뿐아니라 모델의 가중치가 가능한 작게 유지되도록 노력
=>규제항은 훈련하는 동안에만 비용함수에 추가
=>모델의 훈련이 끝나면 모델의 성능을 규제가 없는 성능 지표로 평가
=>입력 특성의 스케일에 민감하기 때문에 수행하기 전에 데이터의 스케일을 맞추는 것이 중요하다.

*라쏘 회귀(least absolute shrinkage and selection operator, Lasso)
=>선형회귀의 또 다른 규제 버전
=>덜 중요한 특성의 가중치를 제거하려고 함 -> 자동으로 특성 선택을 하고 희소모델(spare model)을 만듬
=>라쏘를 사용할 때 경사 하강법이 최적점 근처에서 진동하는 것을 막으려면 훈련하는 동안 점진적으로 학습률을 감소시켜야 합니다.(여전히 최적점 근처에서 진동하겠지만 스텝이 갈수록 작아지므로 수렴하게 될것입니다.)

*엘라스틱넷(elastic net)
=>릿지 회귀와 라쏘 회귀를 절충

*조기 종료(early stopping)
=>검증 에러가 최소값에 도달하면 바로 훈련을 중지 하는 방법

*로지스틱 회귀(logistic regression)
=>샘플이 특정 클래스에 속할 확률을 추정하는데 널리 사용, 추정 확률이 50%가 넘으면 그 샘플이 해당클래스에 속한다고 예측, 아니면 클래스에 속하지 않는 다고 예측 : 이진 분류기
=>시그모이드 함수 == 로지스틱 함수

*소프트 맥스 회귀(다중 로지스틱 회귀)
=>여러 개의 이진 분류기를 훈련시켜 연결하지 않고 직접 다중 클래스를 지원하도록 일반화 하는 방법
=>한번에 하나의 클래스만 예측(다중 출력X)
=>상호 배타적인 클래스에서만 사용해야 함

*수백만 개의 특성을 가진 훈련세트에서는 어떤 선형 회귀 알고리즘을 사용할 수 있을까
=>수백만 개의 특성이 있는 훈련세트를 가지고 있다면 확률적 경사하강법이나 미니배치 경사하강법을 사용할 수 있다. 훈련세트가 메모리 크기에 맞으면 배치경사 하강법도 가능하다. 하지만 정규방정식이나 SVD방법은 계산 복잡도가 특성 개수에 따라 매우 빠르게 증가하기 때문에 사용할 수 없다.

*훈련 세트에 있는 특성들이 각기 아주 다른 스케일을 가지고 있다. 이런 데이터에 잘 작동하지 않는 알고리즘과 그 이유, 해결방법은 무엇인가
=>훈련세트에 있는 특성의 스케일이 매우 다르면 비용함수는 길쭉한 타원 모양의 그릇 형태가 된다. 그래서 경사 하강법 알고리즘이 수렴하는데 오랜 시간이 걸릴것이다. 이를 해결하기 뒤해서는 모델을 훈련하기전에 데이터의 스케일을 조절해야 한다. 정규 방정식이나 SVD방법은 스케일 조정 없이도 잘 작동한다. 또한 규제가 있는 모델은 특성의 스케일이 다르면 지역 최적점에 수렴 할 가능성이 있다. 규제는 가중치가 커지지 못하게 제약을 가하므로 특성값이 작으면 큰 값을 가진 특성에 비해 무시되는 경향이 있다.

*경사 하강법으로 로지스틱 회귀 모델을 훈련시킬 때 지역 최솟값에 갇힐 가능성이 있을까
=>로지스틱 회귀 모델의 비용함수는 볼록 함수이므로 경사 하강법이 훈련될 때 지역 최솟값에 갇힐 가능성 X

*충분히 오랫동안 실행하면 모든 경사 하강법 알고리즘이 같은 모델을 만들어낼까
=>최적화할 함수가 선형회귀나 로지스틱 회귀처럼 볼록함수이고 학습률이 너무 크지 않다고 가정하면 모든 경사 하강법 알고리즘이 전역 최적값에 도달하고 결국 비슷한 모델을 만들것이다. 대신 전역 최적점 주변을 이리저리 매돌게 된다. 결론으로는 매우 오랫동안 훈련해도 경사하강법 알고리즘들은 조금씩 다른 모델을 만들게 된다는 뜻이다.

*배치 경사 하강법을 사용하고 에포크마다 검증 오차를 그래프로 나타내 봤다. 검증 오차가 일정하게 상승하고 있다면 어떤 일이 일어나고 있는 걸까, 어떻게 해결해야 할까
=>에포크마다 검증 에러가 지속적으로 상승한다면 한 가지 가능성은 학습률이 너무 높고 알고리즘이 발산하는 것일지 모른다. 그러나 훈련 에러가 올라가지 않는 다면 모델이 훈련 세트에 과대적합 되어 있는 것이므로 훈련을 멈추어야한다.

*검증 오차가 상승하면 미니배치 경사 하강법을 증시 중단하는 것이 좋은 방법인가
=>무작위성 때문에 확률적 경사 하강법이나 미니배치 경사 하강법은 모두 매 훈련 반복마다 학습의 진전을 보장하지 못한다. 검증 에러가 상승될 때 훈련을 즉시 멈춘다면 최적점에 도달하기 전에 너무 일찍 멈추게 될지 모른다. 그 다음 더 나은 방법은 정기적으로 모델을 정하고 오랫동안 진전이 없을 때(최상의 점수를 넘어서지 못한다면), 저장된 것중 가장 좋은 모델로 복원하는 것이다.

*어떤 경사 하강법 알고리즘이 가장 빠르게 최적 솔루션의 주변에 도달할까, 실제로 수렴하는 것은 어떤 것인가, 다른 방법들도 수렴하게 만들수 있나
=>확률적 경사 하강법 : 한번에 하나의 샘플만 사용하기 때문에 훈련 반복이 가장 빠르다. 그래서 가장 먼저 전역 최적점데 도달한다.(그 다음이 작은 미니배치 크기를 가진 미니배치 경사하강법이다.) 그러나 훈련시간이 충분하면 배치 경사 하강법만 실제로 수렴할 것이다. 앞서 언급한 대로 학습률을 점진적으로 감소시키지 않으면 SGD와 미니배치 GD는 최적점 주변을 맴돌것이다.

*다항 회귀를 사용했을 때 학습 곡선을 보니 훈련오차와 검증오차 사이에 간격이 크다. 무슨 일이 생긴걸까, 어떻게 해결해야 할까
=>검증 오차가 훈련 오차보다 훨씬 더 높으면 모델이 훈련세트에 과대적합 되었을 가능성이 높다. 이를 해결하는 첫번째 방법은 다항 차수를 낮추는 것이다. 자유도를 줄이면 과대적합이 훨씬 줄어들 것이다. 두번째 방법은 모델을 규제하는 것이다. 예를 들면 비용함수에 릿지나 라쏘를 추가한다. 이 방법도 모델의 자유도를 감소시킨다. 세번째 방법은 훈련 세트의 크기를 증가시키는 것이다.

*릿지 회귀를 사용했을 때 훈련 오차와 검증 오차가 거의 비슷하고 둘 다 높았다. 이 모델에는 높은 편향이 문제인가 아니면 높은 분산이 문제인가, 규제 하이퍼파라미터 알파를 증가시켜할까 아니면 줄여야할까
=>훈련 에러와 검증에러가 거의 비슷하고 매우 높다면 모델이 훈련 세트에 과소적합되었을 가능성이 높다. 즉, 높은 편향을 가진 모델이다. 따라서 규제하이퍼 파라미터 알파를 감소시켜야한다.

*다음과 같이 사용해야하는 이유는
1.평범한 선형회귀 대신 릿지 회귀 2.릿지 회귀 대신 라쏘 회귀 3.라쏘 회귀 대신 엘라스틱 넷
=>1.규제가 있는 모델이 일반적으로 규제가 없는 모델보다 성능이 좋다. 그래서 평범한 선형회귀보다 릿지 회귀가 선호된다.
=>2.라쏘 회귀는 가중치를 완전히 0으로 만드는 경향이 있다. 이는 가장 중요한 가중치를 제외하고는 모두 0이 되는 희소한 모델을 만든다. 또한 자동으로 특성 선택의 효과를 가지므로 단지 몇 개의 특성만 실제 유용할 것이라고 의심될 때 사용하면 좋다. 만약 확신이 없다면 릿지 회귀를 사용해야한다.
=>3.라쏘가 어떤 경우(몇 개의 특성이 강하게 연관되어 있거나 훈련 샘플보다 특성이 더 많을 때)에는 불규칙하게 행동하므로 엘라스틱 넷이 라쏘보다 일반적으로 선호된다. 그러나 추가적인 하이퍼파라미터가 생긴다. 불규칙한 행동이 없는 라쏘를 원하면 엘라그틱 넷에 l1_ratio를 1에 가깝게 설정하면된다.

*사진을 낮과 밤, 실내와 실외로 분류하려고 한다. 두 개의 로지스틱 회귀 분류기를 만들어야 할까 아니면 하나의 소프트 맥스 회귀 분류기를 만들어야 할까
=>실외와 실내, 낮과 밤에 따라 사진을 구분하고 싶다면 디 둘은 배타적인 클래스가 아니기 때문에(네가지 조합이 모두 가능하므로) 두개의 로지스틱 회귀 분류기를 훈련 시켜야 한다.

Chapter 5 서포트 벡터 머신

*서포트 벡터 머신(support vector machine, SVM)
=>매우 강력하고 선형이나 비선형 분류, 회귀, 이상치 탐색에도 사용할 수 있는 다목적 머신러닝 모델
=>복잡한 분류 문제에 잘 들어 맞으며 작거나 중간 크기의 데이터셋에 적합
=>SVM은 특성의 스케일에 민감하다.
=>특성의 스케일을 조정하면 결정경계가 훨씬 좋아진다

*라지 마진 분류(large margin classification)
=>SVM 분류기를 클래스 사이에 가장 폭이 넓은 도로를 찾는 것

*서포트 벡터(support vector)
=>도로 경계에 위치한 샘플 위 그림에 동그라미로 표시되는 부분

*하드마진분류(hard margin classification)
=>모든 샘플이 도로 바깥쪽에 올바르게 분류
=>문제점 : 데이터가 선형적으로 구분될 수 있어야 제대로 작동, 이상치에 민감

*소프트마진분류(soft margin classification)
=>도로의 폭을 가능한 넓게 유지하는 것과 마진 오류사이에 적절한 균형을 잡는 방법

*서포트 벡터 머신의 근본 아이디어는
=>SVM의 근본적인 아이디어는 클래스 사이에 가능한 한 가장 넓은 "도로"를 내는 것이다. 다시 말해 두 클래스를 구분하는 결정 경계와 샘플 사이의 마진을 가능한 가장 크게 하는 것이 목적이다. 소프트 마진 분류를 수행할 때는 SVM이 두 클래스를 완벽하게 나누는 것과 가정 넓은 도로를 만드는 것 사이에 절충안을 찾는다.(즉, 몇 개 몇 샘플은 도로 안에 놓일 수 있다.) 또 하나의 핵심적인 아이디어는 비선형 데이터셋에서 훈련할 때 커널 함수를 사용하는 것이다.

*서포트 벡터가 무엇인가
=>SVM이 훈련된 후에 경계를 포함해 도로에 놓인 어떤 샘플이다. 결정 경계는 전적으로 서포트 벡터에 의해 결정된다. 서포트 벡터가 아닌(도로 밖에있는) 어떤 샘플도 영향을 주지 못한다. 이런 샘플은 삭제하고 다른 샘플을 추가하거나, 다른 곳으로 이동시킬 수 있다. 샘플이 도로 밖에 있는 한 결정 경계에 영향을 주지 못할 것이다. 예측을 계산할 때는 전체 훈련세트가 아니라 서포트 벡터만 관여된다.

*SVM을 사용할 때 입력값의 스케일이 왜 중요한가
=>SVM은 클래스 사이에 가능한 한 가장 큰 도로를 내는 것이므로 훈련 세트의스케일이 맞지 않으면 크기가 작은 특성을 무시하는 경향이 있다.

*SVM분류기가 샘플을 분류할 때 신뢰 점수화 확률을 출력할 수 있나
=>SVM분류기는 테스트 샘플과 결정 경계 사이의 거리를 출력할 수 있으므로 이를 신뢰도 점수로 사용할 수 있다. 그러나 이 점수를 클래스 확률의 추정값으로 바로 변환할 수는 없다. 사이킷럿에서 SVM모델을 만들 때 probability = True로 설정하면 훈련이 끝난 후 SVM의 점수에 로지스틱 회귀를 훈련시켜 확률을 계산한다. 이 설정은 SVM모델에 predict_proba()와 predict_log_proba() 메서드를 추가한다.

*수백만 개의 샘플과 수백 개의 특성을 가진 훈련 세트에 SVM모델을 훈련시키려면 원 문제와 쌍대 문제 중 어떤 것을 사용해야할까
=>커널 SVM은 쌍대 형식만 사용할 수 있기 때문에 이 질문은 선형 SVM에만 해당한다. 원 문제의 계산 복잡도는 훈련 샘플 수 m에 비례하지만, 쌍대 형식의 계산 복잡도는 m**2과 m**3사이의 값에 비례한다. 그러므로 수백만개의 샘플이 있다면 쌍대형식은 너무 느려 질것이므로 원 문제를 사용해야한다.

*RBF커널을 사용해 SVM분류기를 훈련시켰더니 훈련 세트에 과소적합된 것 같다. Gamma를 증가시켜야할까 감소시켜야 할까, C의 경우는 어떠나
=>RBF 커널에 훈련된 SVM분류기가 훈련세트에 과소적합이라면 규제가 너무 큰 것일수 있다. 규제를 줄이려면 gamma나 C 또는 둘다 값을 증가 시켜야한다.

*이미 만들어진 QP 알고리즘 라이브러리를 사용해 소프트 마진 선형 SVM 분류기를 학습 시키려면 QP 매개변수(H, f, A, b)를 어떻게 지정해야 하나
=>하드 마진 문제에 대한 QP파라미터를 H',f', A',b'라고 하자. 소프트 마진 문제의 QP파라미터는 m개의 추가적인 파라미터와 m개의 추가적인 제약을 가진다.
=>H 는 H'의 오른쪽에 0으로 채워진 m개의 열이 있고 아래에 0으로 채워진 m개의 열이 있는 행렬
=>f 는 f'에 하이퍼파라미터 C와 동일한 값의 원소 m개가 추가된 벡터
=>b 는 b에 값이 0인 원소 m개가 추가된 벡터
=>A 는 A'의 오른쪽에서 -I(m*m 단위행렬)이 추가되고 바로 그아래 -I이 추가되며 나머지는 0으로 채워진 행렬

Chapter 6 결정 트리

*결정 트리(decision tree)
=>SVM과 유사하게 분류와 회귀 작업 그리고 다중 출력 작업도 가능한 다재 다능한 머신러닝 알고리즘이다. 매우 복잡한 데이터셋도 학습 할 수 있는 강력한 알고리즘

*루트 노드(rood node) : 깊이가 0인 맨 꼭대기의 노드, 시작점
리프 노드(leaf node) : 자식 노드를 가지지 않는 노드, 추가적인 검사를 하지 않는다.

*사이킷런은 이진트리만 만드는 CART알고리즘을 사용한다. 그러므로 리프 노드 외의 모든 노드는 자식 노드를 두개 씩 가진다.

*ID3같은 알고리즘은 둘 이상의 자식 노드를 가진 결정 트리를 만들 수 있다.

*CART(classification and regression tree)
=>결정 트리를 훈련시키기 위해(트리를 성장 시키기 위해 사용)
=>CART 알고리즘이 훈련 세트를 성공적으로 둘로 나누었다면 같은 방식으로 서브셋을 또 나누고 그 다음엔 서브셋의 서브셋을 나누고 이런식으로 반족 -> 최대깊이가 되면 중지하거나 불순도를 줄이는 분할을 찾을 수 없을 때 멈추게 된다.

*특성의 수와 무관하게 계산 복잡도가 같기 때문에 큰 훈련세트를 다룰 때도 예측 속도가 매우 빠르다. 각 노드에서 모든 훈련 샘플의 모든 특성을 비교한다.

*지니 불순도가 가장 빈도 높은 클래스를 한쪽 가지고 고립 시키는 경향이 있는 반면에 엔트로피는 조금 더 균형 잡힌 트리를 만든다.

*비파라미터 모델(nonparameter model) : 훈련되기전에 파라미터 수가 결정 되지 않은 모델
파라미터 모델(parameter model) : 미리 정의된 모델 파라미터 수를 가지므로 자유도가 제한되고 과대 적합 될 위험이 줄어듬(하지만 과서 적합될 위험은 커짐)

*결정 트리는 계단 모양의 결정 경계를 만듬 -> 훈련세트의 회전에 민감
=>결정 트리의 주된 문제는 훈련 데이터에 있는 작은 변화에도 매우 민감하다는 것을 의미

*백만 개의 샘플을 가진 훈련세트에서 (규제 없이) 훈련시킨 결정 트리의 깊이는 대략 얼마 일까
=>m개의 리프 노드를 포함한 균형이 잘 잡힌 이진 트리의 깊이는 log2(m)을 반올림 한 것과 같다. 이진 결정 트리(사이킷런에 있는 모든 트리는 가지가 두개이다) 를 제한을 두지 않고 훈련시키면 훈련 샘플마다 하나의 리프 노드가 되므로 어느 정도 균형이 잡힌 트리가 된다. 따라서 훈련 세트에 백만 개 샘플이 있다면 결정 트리의 깊이는 log2(10**6) ~~ 20 이 될것이다.(실제로는 조금 더 늘어남)

*한 노드의 지니 불순도가 보통 그 부모 노드보다 작을까요, 클까요, 일반적으로 작거나 클까요 아니면 항상 작거나 클까요
=>한 노드의 불순도는 일반적으로 부모의 불순도보다 낮다. 이는 자식의 지니 불순도의 가중치 합이 최소화 되는 방향으로 각 노드를 분할하는 CART훈련 알고리즘의 비용함수 때문이다. 그러나 다른 자식 노드의 지니 불순도 감소량이 어떤 노드의 불순도 증가량보다 큰 경우라면 부모의 불순도보다 큰 노드가 생길 수 있다.

*결정 트리가 훈련 세트에 과대적합 되었다면 max_depth를 줄이는 것이 좋을까
=>결정 트리가 훈련 세트에 과대 적합 되었다면 모델에 제약을 가해 규제해야 하므로 max_depth를 낮추는 것이 좋다.

*결정 트리가 훈련 세트에 과소 적합 되었다면 입력 특성의 스케일을 조정하는 것이 좋을까
=>결정 트리는 훈련 데이터의 스케일이나 원점에 맞추어져 있는지 상관하지 않는다. 이것이 결정 트리의 장점 중 하나이다. 그러므로 결정 트리가 훈련 세트에 과소적합 되었다고 입력 특성의 스케일을 조정하는 것은 시간 낭비이다.

*백만개의 샘플을 가진 훈련 세트에 결정 트리를 훈련시키는데 한시간이 걸렸다면, 천만개의 샘플을 가진 훈련 셑에 결정 트리를 훈련시키는 데는 대략 얼마나 걸릴까
=>결정 트리 훈련의 계산 복잡도는 O(n*m*log2(m))이다. 그러므로 훈련 세트의 크기에 10을 곱하면 10*log2(10m)/log2(m)배 만큼 늘어난다. 만약 m=10**6이면 대락 11.7시간이 걸릴 것으로 예상된다.

*십만 개의 샘플을 가진 훈련 세트가 있다면 presort=True로 지정하는 것이 훈련속도를 높일까
=>데이터셋의 샘플 수가 수천 개 미만일 때 훈련 세트를 사전에 정렬하여 훈련 속도를 높일 수 있다. 100,000개의 샘플을 포함하고 있을 때 presort=True로 지정하면 훈련 속도가 매우 느려질 것이다.

Chapter 7 앙상블 학습과 랜덤 포레스트

*앙상블 학습
=>일련의 예측기(분류나 회귀모델)로부터 예측을 수집하면 가장 좋은 모델 하나보다 더 좋은 예측을 얻을 수 있을 것이다.

*앙상블 방법
=>훈련세트로부터 무작위로 각기 다른 서브셋을 만들어 일련의 결정 트리 분류기를 훈련 시킬 수 있다. 예측을 하려면 모든 개별 트리의 예측을 구하면 되고 그런 다음 가장 많은 선택을 받은 클래스를 예측으로 삼는다. 결정 트리의 앗앙블을 랜덤 포레스트(random forest)라고 한다.

*직접 투표(hard voting)
=>더 좋은 분류기를 만드는 매우 간단한 방법은 각 분류기의 예측을 모아서 가장 많이 선택된 클래스를 예측하는 것이다. 각 분류기는 약한 학습기(weak learner)일지라도 충분하게 많고 다양하다면 앙상블은 강한 학습기(strong learner)가 될 것이다.

*앙상블 방법은 예측기가 가능한 한 서로 독립적일 때 최고의 성능을 발휘한다. 다양한 분류기를 얻는 한가지 방법은 각기 다른 알고리즘으로 학습시키는 것이다. 이렇게 하면 매우 다른 종류의 오차를 만들 가능성이 높기 때문에 앙상블 모델의 정확도를 향상 시킨다.

*간접 투표(soft voting)
=>모든 분류기가 클래스의 확률을 예측할수 있으면, 개별 분류기의 예측을 평균 내어 가장 높은 클래스를 예측할 수 있다. 이 방식은 확률이 높은 투표에 비중을 더 두기 때문에 직접 투표 방식보다 성능이 높다.

*배깅(bagging) : 훈련세트에 중복을 허용하여 샘플링 하는 방식
페이스팅(pasting) : 중복을 허용하지 않고 샘플링 하는 방식

*앙상블은 비슷한 편향에서 더 작은 분산을 만든다.(훈련 세트의 오차 수가 거의 비슷하지만 결정 경계는 덜 불규칙하다)

*특성 샘플링
=>예측기는 무작위로 선택한 입력 특성의 일부분으로 훈련, 이미지와 같은 매우 고차원의 데이터셋을 다룰 때 유용, 더 다양한 예측기를 만들며 편향을 늘리는 대신 분산을 낮춘다.

*랜덤 패치 방식(random patches method) : 훈련 특성과 샘플을 모두 샘플링하는 것
랜던 서브스페이스 방식(random subspace method) : 훈련 샘플을 모두 사용하고 특성은 샘플링 하는 것

*랜덤 포레스트(random forest)
=>일반적으로 배깅 방법(또는 페이스팅)을 적용한 결정 트리의 앙상블
=>트리의 노드를 분할 할 때 전체 특성 중에서 최선의 특성을 찾는 대신 무작위로 선택한 특성 후보 중에서 최적의 특성을 찾는 식으로 무작위성을 더 주입한다. 트리를 더욱 다양하게 만들고 편향을 손해보는 대신 분산을 더 낮추어 훌륭한 모델을 만들게 된다.

*익스트림 랜덤 트리(extremely randomized tree)
=>트리를 더욱 무작위로 만들기 위해 최적의 임곗값을 찾는 대신 후보 특성을 사용해 무작위로 분할한 다음 그 중에서 최상의 분할을 선택한다. 극단적으로 무작위한 트리의 랜덤 포레스트, 편향은 늘고 분산은 낮춤, 일반적이 랜덤포레스트 보다 엑스트라 트리가 훨씬 빠르다

*특성 중요도
=>사이킷런은 어떤 특성을 사용한 노드가 평균적으로 분수도를 얼마나 감소시키는지 확인하여 특성의 중요도를 측정한다. 가중치 평균이며 각 노드의 가중치는 연관된 훈련 샘플 수와 같다. 사이킷런은 훈련이 끝난 뒤 특성마다 자동으로 이 점수를 계산하고 중요도의 전체합이 1이 되도록 결과값을 정규화 한다.

*부스팅(boosting)
=>약한 학습기를 여러 개 연결하여 학습기를 만드는 앙상블 방법,
=>아이디어 : 모델을 보완해 나가면서 일련의 예측기를 학습시키는 것

*에이다 부스트
=>이전 모델이 과소적합했던 훈련 샘플의 가중치를 더 높이는 것이다. 새로운 예측기는 학습하기 어려운 샘플에 점점 더 맞춰지게 된다.

*연속된 학습 기법에는 중요한 단점이 하나 있다. 각 예측기는 이전 예측기가 훈련되고 평가 된 후에 학습 될 수 있기 때문에 병렬화(또는 분할)를 할 수 없다. 결국 배깅이나 페이스팅만큼 확장성이 높지 않다.

*그레이디언트 부스팅(gradient boosting)
=>에이다 부스트처럼 앙상블에 이전까지의 오차를 보정하도록 예측기를 순차적으로 추가한다. 에이다 부스트처럼 반복마다 샘플의 가중치를 수정하는 대신 이전 예측기가 만든 잔여 오차(residual error)에 새로운 예측기를 학습 시킨다.

*조기종료
=>최적의 트리수를 찾기 위한 방법, 훈련의 가 단계에서 앙상블에 의해 만들어진 예측기를 순회하는 반복자(iterator)를 반환한다.

*확률적 그레이디언트 부스팅(stochastic gradient boosting)
=>각 트리가 훈련할 때 사용할 훈련 샘플의 비율을 지정하여 학습한다. 편향이 높아지는 대신 분산이 낮아지게 된다. 훈련속도가 높다.

*XGBoost(익스트림 그레이디언트 부스팅, extreme gradient boosting)
=>목표 : 매우 빠른 속도, 확장성, 이식성
=>머신러닝 경연 대회에서 우승 후보들이 사용하는 중요 도구 중 하나

*스태킹(stacking, stacked generalization)
=>앙상블에 속한 모든 예측기의 예측을 취합하는 간단한 함수(직접 투표 같은)를 사용하는 대신 취한하는 모델을 훈련시킬수는 없을까? 라는 아이디어로부터 나옴

*정확히 같은 훈련 데이터로 다섯개의 다른 모델을 훈련시켜서 모두 95%의 정확도를 얻는다면 이 모델들을 연결하여 더 좋은 결과를 얻을 수 있을까, 가능하다면 어떻게 해야 할까, 그렇지 않다면 왜 그럴까
=>다섯개의 모델을 훈련 시켰고 모두 95%의 정확도를 달성했다면 이들을 연결하여 투표 앙상블(voting ensemble)을 만들어 더 나은 결과를 기대 할 수 있다. 만약 모델이 서로 다르다면(예를 들면 SVM분류기, 결정트리 분류기, 로지스틱 회귀 분류기 등)훨씬 좋다. 만약 다른 훈련 샘플에서 훈련 되었다면 더더욱 좋다.(이것이 배깅과 페이스팅 앙상블의 핵심). 하지만 그렇지 않더라고 모델이 서로 많이 다르면 여전히 좋은 결과를 낸다.

*직접 투표와 간접 투표 분류기 사이의 차이점은 무엇일까
=>직접 투표 분류기는 앙상블에 있는 각 분류기의 선택을 분류기의 선택을 카운트해서 가장 많은 투표를 얻는 클래스를 선택한다. 간접 투표 분류기는 각 클래스의 평균적인 확률 추정값을 계산해서 가장 높은 확률을 가진 클래스를 고른다. 이 방식은 신뢰가 높은 투표에 더 가중치를 주고 종종 더 나은 성능을 낸다. 하지만 앙상블에 있는 모든 분류기가 클래스 확률을 추정할 수 있어야 사용할 수 있다.

*배깅 앙상블의 훈련을 여러 대의 서버에 분산시켜 속도를 높일 수 있을까, 페이스팅 앙상블, 부스팅 앙상블, 랜덤 포레스트, 스태킹 앙상블의 경우는 어떨까
=>배깅 앙상블의 각 예측기는 독립적이므로 여러대의 서버에 분산하여 앙상블의 훈련 속도를 높일 수 있다. 페이스팅 앙상블과 랜덤 포레스트도 같은 이유로 동일하다. 그러나 부스팅 앙상블의 예측기는 이전 예측기를 기반으로 만들어지므로 훈련이 순차적이어야하고 여러 oe의 서버에 분산해서 얻을 수 있는 이득이 없다. 스태킹 앙상블의 경우 한 층의 모든 예측기가 각각 독립적이므로 여러대의 서버에서 병렬로 훈련 될 수 있다. 그러나 한 층에 있는 예측기들은 이전 층의 예측기들이 훈련된 후에 훈련 될 수 있다.

*oob평가의 장점은 무엇인가
=>oob 평가를 사용하면 배깅 앙상블의 각 예측기가 훈련에 포함되지 않은(따로 떼어 놓은) 샘플을 사용해 평가된다. 이는 추가적인 검증 세트가 없어도 편향되지 않게 앙상블을 평가하도록 도와준다. 그러므로 훈련에 더 많은 샘플을 사용할 수 있어서 앙상블의 성능은 조금 더 향상될 것이다.

*무엇이 엑스트라 트리를 일반 랜덤 포레스트 보다 더 무작위 하게 만드나, 추가적인 무작위성이 어떻게 도움이 될까, 엑스트라 트리는 일반 랜덤 포레스트 보다 느릴까 빠를까
=>랜덤 포레스트에서 트리가 성장할 때 각 노드에서 특성의 일부를 무작위로 선택해 분할에 사용한다. 엑스트라 트레이서도 이는 마찬가지지만 한 단계 더 나아가서 일반 결정 트리처럼 가능한 최선의 임계점을 찾는 것이 아니라 각 특성에 대해 랜덤한 임계점을 사용한다. 이 추가적인 무작위성은 규제처럼 작동한다. 즉, 랜덤 포레스트가 훈련 데이터에 과대 적합되었다면 엑스트라 트리는 그렇지 않을 것이다. 또한 엑스트라 트리는 가능한 최선의 임계점을 찾지 않기 때문에 랜덤 포레스트보다 훨씬 빠르게 훈련된다. 그러나 예측을 할 때는 랜덤포레스트보다 더 빠르지도 느리지도 않다.

*에이다 부스트 앙상블이 훈련 데이터에 과소적합 되었다면 어떤 매개변수를 어떻게 바꾸어야 할까
=>에이더 부스트 앙상블이 훈련 데이터에 과소적합 되었다면 예측기 수를 증가 시키거나 기반 예측기의 규제 하이퍼파라미터를 감소시켜 볼 수 있다. 또한 학습률을 약간 증가 시켜 볼 수 있다.

*그레이디언트 부스팅 앙상블이 훈련 데이터에 과대적합되었다면 학습률을 높여야할까 낮춰야할까
=>그레이디언트 부스팅 앙상블이 훈련 세트에 과대적합되었다면 학습률을 감소 시켜야한다. (예측기수가 너무 많으면) 알맞은 개수를 찾기 위해 조기 종료 기법을 사용 할 수 있다.

Chapter 8 차원 축소

*매니폴드
=>d차원 매니폴드는 국부적으로 d차원 초평면으로 보일 수 있는 n차원 공간의 일부이다. 많은 차원 축소 알고리즘이 훈련 샘플이 놓여있는 매니폴드를 모델링하는 식으로 작동한다. 이를 매니폴드 학습(manifold learning)이라고 한다. 이는 대부분 실제고차원 데이터셋이 더 낮은 저차원 매니폴드에 가깝게 놓여 있다는 매니폴드 가정 또는 매니폴드 가설에 근거한다.

*PCA(principal component analysis, 주성분 분석)
=>가장 인기 있는 차원 축소 알고리즘, 먼저 데이터에 가까운 초 평면을 정의한 다음에 데이터를 이 평면에 투영시킨다.
=>각 주성분을 위해 PCA는 주성분 방향을 가리키고 원점에 중앙이 맞춰진 단위 벡터를 찾는다. 하나의 축에 단위 벡터가 반대 방향으로 두개 있으므로 PCA가 반환하는 단위 벡터의 방향은 일정하지 않다. 훈련세트를 조금 섞은 다음 다시 PCA를 적용하면 새로운 PC중 일부가 원래 PC와 반대 방향일 수 있다. 그러나 일반적으로 같은 축에 놓여 있을 것이다. 어떤 경우에는 한쌍의 PC가 회전하거나서로 바꿀 수 있지만 보통은 같은 평면을 구상한다.

*설명된 분산의 비율(explained variance ratio) : 각 주성분의 축을 따라 데이터셋의 분산 비율을 나타냄

*랜덤 PCA
=>확률적 알고리즘을 사용해 처음 d개의 주성분에 대한 근삿값을 빠르게 찾는다. d가 n보다 많이 작으면 완전 SVD 보다 훨씬 빠르다.

*점진적 PCA(incremental PCA, IPCA)
=>PCA 구현의 문제는 SVD 알고리즘을 실행하기 위해 전체 훈련세트를 메모리에 올려야 한다는 문제점을 보완하기 위한 알고리즘, 훈련세트를 미니배치로 나눈 뒤 IPC알고리즘에 한번에 하나씩 주입, 훈련세트가 클 때 유용하고 온라인으로 PCA를 적용 할 수도 있다.

*커널 PCA(kPCA)
=>차원 축소를 위한 복잡한 비선형 투영을 수행한다. 투영된 후에 샘플의 군집을 유지하거나 꼬인 매니폴드에 가까운 데이터셋을 펼칠 때 유용하다.
=>비지도 학습이기 때문에 좋은 커널과 하이퍼 파라미터를 선택하기 위한 명확한 성능 측정 기준이 없다. 하지만 차원 축소는 종종 지도 학습의 전처리 단계가 활용되므로 그리드 탐색을 사용하여 주어진 문제에서 성능이 가장 좋은 커널과 하이퍼파라미터를 선택할 수 있다.

*재구성 원상(pre-image)
=>재구성된 포인트에 가깝게 매핑 된 원본 공간의 포인트, 원상을 얻게 되면 원본 샘플과의 제곱 거리를 측정할 수 있다. 투영된 샘플을 훈련세트로, 원본 샘플을 타깃으로 하는 지도 학습 회귀 모델 훈련

*LLE(locally linear embedding, 지역 선형 임베딩) == NLDR(nonlinear dimensionlity reduction, 비선형 차원 축소)
=>투영에 의존하지 않는 매니폴드 학습, 먼저 각 훈련 샘플이 가장 가까운 이웃에 얼마나 선형적으로 연관도었는지 측정한 후 국부적인 관계가 가장 잘 보존되는 훈련 세트의 저차원 표현을 찾는다. 잡음이 너무 많지 않은 경우 꼬인 매니폴드를 펼치는데 잘 작동한다.

*랜덤 투영(random projection)
=>랜덤한 선형 투영을 사용해 데이터 저차원 공간으로 투영한다. 실제로 거리를 잘 보조한다.

*다차원 스케일링(multidimensional scaling, MDS)
=>샘플 간의 거리를 보존하면서 차원을 축소

*Isomap
=>각 샘플을 가장 가까운 이웃과 연결하는 식으로 그래프를 만든다. 그런 다음 샘플 간의 지오데식 거리(geodesic distance)를 유지하면서 차원을 축소한다.

*t-SNE(t-distributes stochastic neighbor embedding)
=>비슷한 샘플은 가까이, 비슷하지 않은 샘플은 멀리 떨어지도록 하면서 차원축소, 주로 시각화에 많이 사용되면 특히 고차원 공간에 있는 샘플의 군집을 시각화 할 때 사용된다.

*선형 판별 분석(linear discriminant analysis, LDA)
=>사실 분류 알고리즘, 훈련 과정에서 클래스 사이를 가장 잘 구분하는 축을 학습, 이 축은 데이터가 투영되는 초평면을 정의하는데 사용할 수 있다.
=>장점 : 투영을 통해 가능한 클래스를 멀리 떨어지게 유지 시키므로 SVM분류기 같은 다른 분류 알고리즘을 적용하기 전에 차원을 축소 시키는 데 좋다.

*데이터셋의 차원을 축소하는 주요 목적은 무엇인가, 대표적인 단점은 무엇인가
=>목적 : 훈련 알고리즘의 속도를 높이기 위해(어떤 경우에는 잡음과 중복된 특성을 삭제할 수도 있어 훈련 알고리즘의 성능을 높임), 데이터를 시각화 하고 가장 중요한 특성에 대한 통찰을 얻기 위해, 메모리 공간을 절약하기 위해(압축)
=>단점 : 일부 정보를 잃어버려 훈련 알고리즘의 성능을 감소시킬 수 있다. 계산 비용이 높다. 머신러닝 파이프라인의 복잡도를 증가 시킨다. 변환된 데이터를 이해하기 어려운 경우가 많다.

*차원의 저주란 무엇인가
=>저차원의 공간에 없는 많은 문제가 고차원 공간에서 일어난다는 사실을 뜻한다. 머신러닝에서 무작위로 선택한 고차원 벡터는 매우 희소해서 과대적합의 위험이 크고, 많은 양의 데이터가 있지 않으면 데이터에 있는 패턴을 잡아내기 어려운 것이 흔한 현상이다.

*데이터셋의 차원을 축소시키고 나서 이 작업을 원복할수 있나요, 할 수 있다면 어떻게 가능할까, 가능하지 않다면 왜일까
=>여기에서 설명한 알고리즘 중 하나를 사용해 데이터셋의 차원이 축소되면 일부 정보가 차원 축소 과정에서 사라지기 때문에 이를 완벽하게 되돌리는 것은 불가능하다. (PCA 같은) 일부 알고리즘은 비교적 원본과 비슷한 데이터셋을 재구성할 수 있는 간단한 역변환 방법을 가지고 있지만 (T-SNE 같은) 다른 알고리즘들은 그렇지 않다.

*매우 비선형적인 데이터셋의 차원을 축소하는데 PCA를 사용할 수있을까
=>PCA는 불필요한 차원을 제거할 수 있기 때문에 매우 비선형적이더라도 대부분의 데이터셋에서 차원을 축소하는데 사용할 수 있다. 그러나 불필요한 차원이 없다면 (예를 들면 스위스롤 데이터셋)PCA의 차원 축소는 너무 많은 정보를 잃게 만든다. 스위스롤을 펼쳐야하며 말려진 것을 뭉게면 안된다.

*설명된 분산율 95%로 지정한 PCA를 1000개의 차원을가진 데이터셋에 적용한다고 가정하자. 결과 데이터셋의 차원은 얼마나 될까
=>데이터셋에 따라 다르다. 차원 수에 대한 함수로 설명된 분산의 그래프를 그려보는 것이 데이터셋에 내재된 차원수를 대략 가늠할 수 있는 한가지 방법이다

*기본 PCA, 점진적 PCA, 랜덤 PCA, 커널 PCA는 어느 경우에 사용될까
=>기본 PCA가 우선적으로 사용되지만 데이터셋 크기가 메모리에 맞을 때에 가능하다. 점진적 PCA는 메모리에 담을 수 없는 대용량 데이터셋에 적합하다. 하지만 PCA보다 느리므로 데이터셋이 메모리 크기에 맞으면 기본 PCA를사용해야한다. 점진적 PCA는 새로운 샘플이 발생될 때 마다 실시간으로 PCA를 적용해야 하는 온라인 작업에 사용 가능하다. 랜덤 PCA는 데이터셋이 메모리 크기에 맞고 차원을 크게 축소시킬 때 사용된다. 이 경우에는 기본 PCA보다 훨씬 빠르다. 커널 PCA는 비선형 데이터셋에 유용하다.

*어떤 데이터셋에 적용한 차원 축소 알고리즘의 성능을 어떻게 평가할 수 있나
=>직관적으로 데이터셋에 너무 많은 정보를 잃지 않고 차원을 많이 제거할 수 있다면 차원 축소 알고리즘이 잘 작동한 것이다. 이를 측정하는 한 가지 방법은 역변환을 수행해서 재구성 오차를 측정하는 것이다. 하지만 모든 차원 축소를 다른 머신러닝 알고리즘을 적용하기전에 전처리 단계로 사용한다면 두번째 알고리즘의 성능을 측정해 볼 수 있다. 차원 축소가 너무 많은 정보를 잃지 않았다면 원본 데이터셋을 사용했을 때와 비슷한 성능이 나와야 한다.

*두개의 차원 축소 알고리즘을 연결할 수 있을까
=>당연히 두개의 차원 축소 알고리즘을 연결할 수 있다. PCA로 불필요한 차원을 대폭 제거하고 난 다음 LLE 같이 훨씬 느린 알고리즘을 적용하는 것이 대표적인 사례이다. 이런 2단계 방식은 LLE만 사용했을때와 거의 비슷한 성능을 내지만 속도가 몇분의 1로 줄어들 것이다.

Chapter 9 비지도 학습

*군집(clustering)
=>비슷한 심플을 클러스터(cluster)로 모은다. 군집은 데이터분석, 고객분류, 추진시스템, 검색 엔진, 이미지 분할, 준지도 학습, 차원 축소 등에 사용할 수 있는 훌륭한 도구이다.

*이상치 탐지(oulier detection)
=>'정상' 데이터가 어떻게 보이는지를 학습한다. 그 다음 비정상 샘플을 감지하는데 사용한다. 예를 들면 제조 라인에서 결함 제품을 감지하거나 시계열 데이터에서 새로운 트랜드를 찾는다.

*밀도 추정(density estimation)
=>데이터셋 생성활률 과정(random process)의 활률 밀도 함수(PDF)를 추정한다. 밀도 추정은 이상치 탐지에 널리 사용된다. 밀도가 매우 낮은 영역에 놓인 샘플이 이상치일 가능성이 높다. 또한 데이터 분석과 시각화에 유용하다.

*군집(clustering) : 비슷한 샘플을 구별해 하나의 클러스터 또는 비슷한 샘플의 그룹으로 할당하는 작업이다.

*군집을 사용하는 애플리케이션
=>고객분류 : 고객을 구매 이력이나 웹사이트 내 행동 등을 기반으로 클러스터를 모을 수 있다. 이는 고객이 누구인지, 고객이 무엇을 원하는지 이해하는데 도움이 된다. 고객 그룹마다 제품 추천이나 마케팅 전력을 다르게 적용할 수 있다. 예를 들어 동일한 클러스터 내의 사용자가 좋아하는 컨텐츠를 추천하는 추천시스템(recommender system)을 만들 수 있다.
=>데이터분석 : 새로운 데이터셋을 분석 할 때 군집 알고리즘을 실행하고 각 클러스터를 따로 분석하면 도움이 된다.
=>차원 축소 기법 : 한 데이터셋에 군집 알고리즘을 적용하면 각 클러스터에 대한 샘플의 친화성(affinity)를 측정할 수 있다. 각 샘플의 특성 벡터 x는 클러스터 친화성의 벡터로 바꿀 수 있다. k개의 클러스터가 있다면 이 벡터는 k 차원이 된다. 이 벡터는 일반적으로 원본 특성보다 훨씬 더 저차원이다. 하지만 이 후 분석을 위한 충분한 정보를 가질 수 있다.
=>이상치 탐지 : 모든 클러스터에 친화성이 낮은 샘플은 이상치일 가능성이 높다. 웹사이트 내 행동을 기반으로 사용자의 클러스터를 만들었다면 초당 웹서버 요청을 비정상적으로 많이 하는 사용자를 감지 할 수 있따. 이상치 탐지는 특히 제조 분야에서 결함을 감지할 때 유용하다. 또는 부정거래 탐지에 활용된다.
=>준지도 학습 : 레이블된 샘플이 적다면 군집을 수행하고 동일한 클러스터에 있는 모든 샘플에 레이블을 전파할 수 있다. 이 기법은 이어지는 지도 학습 알고리즘에 필요한 레이블이 크게 증가해 성능을 크게 향상한다.
=>검색 엔진 : 일부 검색 엔진은 제시된 이미지와 비슷한 이미지를 찾아준다. 이런 시스템을 구축하려면 먼저 데이터 베이스에 있는 모든 이미지에 군집 알고리즘을 적용해야 한다. 비슷한 이미지는 동일한 클러스터에 속한다. 사용자가 찾으려는 이미지를 제공하면 훈련된 군집 모델을 사용해 이미지의 클러스터를 찾는다. 그 다음 이 클러스터의 모든 이미지를 반환한다.
=>이미지 분할 : 색을 기반으로 픽셀을 클러스터로 모은다. 그 다음 각 픽셀의 색을 해당 클러스터의 평균 색으로 바꾼다. 이는 이미지에 있는 색상의 종류를 크게 줄인다. 이렇게 하면 물체의 윤곽을 감지하기 쉬워져 물체 탐지 및 추적 시스템에서 이미지 분할을 많이 활용한다.

*k-평균
=>반복 몇번으로 이런 종류의 데이터셋을 빠르고 효율적으로 클러스터로 묶을 수 있는 간단한 알고리즘

*센트로이드(centroid) : 특정 포인트

*k-평균 알고리즘
=>처음에 센트로이드를 랜덤하게 초기화, 그다음 센트로이드를 업데이트, 샘플에 다시 레이블을 할당

*이너셔(inertia)
=>성능 지표, 각 샘플과 가장 가까운 샘플과 가장 가까운 센트로이드 사이의 제곱 거리 합

*미니배치 k-평균이 일반 k-평균보다 훨씬 빠르다

*실루엣 점수(silhouette score)
=>최선의 클러스터 개수를 선택하는 방법, 계산 비용이 많이 든다. 모든 샘플에 대한 실루엣 계수의 평균
=>샘플의 실루엣 계수 = (b-a)/max(a,b)
a=동일한 클러스터에 있는 다른 샘플까지 평균거리(클러스터 내부의 평균 거리)
b=가장 가까운 클러스터까지 평균거리(가장 가까운 클러스터의 샘플까지 평균거리, 샘플과 가장 가까운 클러스터는 자신이 속한 클러스터는 제외하고 b가 최소인 클러스터)

*실루엣 다이어그램(silhouette diagram)
=>모든 샘플의 실루엣 계수를 할당된 클러스터와 계수값으로 정렬하여 그리면 더 많은 정보가 있는 그래프를 얻을 수 있다. 그래프의 높이는 클러스터가 포함하고 있는 샘플의 개수를 의미하고 너비는 클러스터에 포함된 샘플의 정렬된 실루엣 계수를 나타낸다. (넓을수록 좋음)

*k-평균
=>장점 : 속도가 빠르고 확장이 용이
=>단점 : 최적이 아닌 솔루션을 피하려면 알고리즘을 여러 번 실행해야 한다. 클러스터 개수를 지정해야 한다. 클러스터의 크기나 밀집도가 서로 다르거나 원형이 아닐 경우 잘 작동 하지 않는다.

*k-평균을 실행하기전에 입력 특성의 스케일을 맞추는 것이 중요하다. 그렇지 않으면 클러스터가 길쭉 해지고 k-평균의 결과가 좋지 않다. 특성의 스케일을 맞추어도 모든 클러스터가 잘 구분되고 원형의 형태를 가진다고 보장할 수는 없지만 일반적으로 더 좋아진다.

*이미지 분할(image segmentation)
=>이미지 세그먼트(segment) 여러 개로 분할 하는 작업

*시맨틱 분할(semantic segmentation)
=>동일한 종류의 물체에 속한 모든 픽셀은 같은 세그먼트에 할당 된다.

*색상 분할(color segmentation)
=>동일한 색상을 가진 픽셀을 같은 세그먼트에 할당

*준지도 학습 : 레이블이 없는 데이터가 많고 레이블이 있는 데이터는 적을 때 사용

*대표 이미지(reprentative image) : 클러스터에서 센트로이드에 가장 가까운 이미지 찾기

*능동학습(active learning)
=>전문가가 학습 알고리즘과 상호작용하여 알고리즘이 요청할때 특정 샘플의 레이블 제공
=>불확실성 샘플링(uncertainty sampling) : 1. 지금까지 수집한 레이블된 샘플에서 모델을 훈련, 이 모델을 사용해 레이블 되지 않은 모든 샘플에 대한 예측을 만든다. 2. 모델이 가장 불확실하게 예측한 샘플(주 추정 확률이 낮은 샘플)을 전문가에게 보내 레이블을 붙인다. 3. 레이블을 부여하는 노력만큼의 성능이 향상되지 않을 때 까지 이를 반복

*DMSCAN
=> 알고리즘이 각 샘플에서 작은 거리인 앱실론 내에 샘플이 몇개 놓여있는지 센다. 이 지역을 샘플의 앱실론-이웃 이라고 부른다. (자기자신을 포함해) 앱실론- 이웃 내에 적어도 min_samples개 샘플이 있다면 핵심샘플(core instance)로 간주, 즉 핵심 샘플은 밀집된 지역에 있는 샘플이다. 핵심 샘플의 이웃에 있는 모든 샘플은 동일한 클러스터에 속한다. 이웃에는 다른 핵심 샘플이 포함될 수 있다. 따라서 핵심 샘플의 이웃의 이웃은 계속해서 하나의 클러스터를 형성한다. 핵심 샘플이 아니고 이웃도 아닌 샘플은 이상치로 판단

*병합 군집(agglomerative clustering)
=>대규모 샘플과 클러스터에 잘 확장되며 다양한 형태의 클러스터를 감지할 수 있다. 특정 클러스터 개수를 선택하는데 도움이 되는 유용한 클러스터 트리를 만들 수 있다. 짝 거리(pairwise distance)와도 사용할 수 있다. 연결 행렬이 없으면 대규모 데이터셋으로 확장하기 어렵다.

*BIRCH(balanced iterative reducing and clustering using hierarchies)
=>특별히 대규모 데이터셋을 위해 고안, 특성개수가 많지 않다면 배치 k-평균보다 빠르고 비슷한 결과를 만든다. 훈련과정에서 새로운 샘플을 클러스터에 빠르게 할당할 수 있는 정보를 담은 트리 구조를 만든다. 제한된 메모리를 사용해 대용량 데이터를 다룰 수 있다.

*평균-이동
=> 지역의 최대 밀도를 찾을 때까지 높은 쪽으로 원을 이동, 모양이나 개수에 상관없이 클러스터를 찾을 수 있다. 하이퍼파라미터도 매우 적다. 국부적인 밀집도 추정의 의존, 내부 밀집도가 불균형할 때 여러 개로 나누는 경향이 있다. 대규모 데이터셋에는 적합X

*유사도 전파
=> 투표방식을 사용, 샘플은 자신을 대표할 수 있는 비슷한 샘플에 투표한다. 알고리즘이 수렴하면 각 대표와 투표한 샘플이 클러스터를 형성, 유사도 전파(affinity propagation)는 크기가 다른 여러 개의 클러스터를 감지할 수 있다. 대규모 데이터셋에는 적합하지 않다.

*스펙트럼 군집
=> 샘플 사이의 유사도 행렬을 받아 저차원 임베딩을 만든다.(차원을 축소한다.) 저차원 공간에서 또 다른 군집 알고리즘을 사용한다. 복잡한 클러스터 구조를 감지하고 그래프 컷(graph cut)을 찾는데 사용할 수 있다. 샘플 개수가 많으면 잘 적용되지 않고 클러스터의 크기가 매우 다르면 잘 작동하지 않는다.

*가우시안 혼합 모델(Gaussian mixture model, GMM)
=>샘플이 파라미터가 알려지지 않은 여러 개의 혼합된 가우시안 분포에서 생성 되었다고 가정하는 확률 모델, 하나의 가우시안 분포에서 생성된 모든 샘플은 하나의 클러스터를 형성(일반적으로 이 클러스터는 타원형)

*이상치 탐지(outlier detection)
=>보통과 많이 다른 샘플을 감지하는 작업

*가우시안 혼합 모델은 이상치를 포함해 모든 데이터에 맞추려고 한다. 따라서 이상치가 너무 많으면 모델이 정상치를 바라보는 시각이 편향되고 일부 이상치를 정상으로 잘못 생각할 수 있다. 이런 일이 일어나면 먼저 한 모델을 훈련하고 가장 크게 벗어난 이상치를 제거한다. 그 다음 정제된 데이터셋에 모델을 다시 훈련한다. 또 다른 방법은 안정적인 공분한 추정 방법을 사용하는 것이다.

*베이즈 가우시안 혼합모델
=>최적의 클러스터 개수를 수동으로 찾지 않고 불필요한 클러스터의 가중치를 0 으로 또는 0 에 가깝게 만드는 모델, 자동으로 불필요한 클러스터를 제거

*베타 분포(beata distribution)
=>고정 범위 안에 놓인 값을 가진 확률 변수를 모델링 할 때 자주 사용, 범위 0~1

*블랙 박스 확률적 변분 추론(black box stochastic variationl inference, BBSVI)
=>각 반복에서 몇 개의 샘플을 q 에서 뽑아 변분 파라미터에 대한 ELBO 의 그레이디언트를 추정하는데 사용

*이상치 탐지와 특이치 탐지를 위한 다른 알고리즘
=>PCA : 보통 샘플의 재구성 오차와 이상치의 재구성 오차를 비교하면 일반적으로 후자가 훨씬 크다. 이는 간단하고 종종 매우 효과적인 이상치 탐지 기법이다.
=>Fast_MCD : EllipticEnvelope 클래스에 구현된 이 알고리즘은 이상치 감지에 유용하다. 특히 데이터셋을 정제할 때 사용된다. 보통 샘플(정상치)가 (혼합된 것이 아니라) 하나의 가우시안 분포에서 생성되었다고 가정한다. 또한 이 가우시안 분포에서 생성되지 않은 이상치로 이 데이터셋이 오염되었다고 가정한다. 알고리즘이 가우시안 분포의 파라미터를 (정상치를 둘러싼 타원 도형을) 추정할 때 이상치로 의심되는 샘플을 무시한다. 이런 기법은 알고리즘이 타원형을 잘 추정하고 이상치를 잘 구분하도록 돕는다.
=>아이솔레이션 포레스트 : 특히 고차원 데이터셋에서 이상치 감지를 위한 효율적인 알고리즘이다. 이 알고리즘은 무작위로 성장한 결정 트리로 구성된 랜덤 포레스트를 만든다. 각 노드에서 특성을 랜덤하게 선택한 다으(최솟값과 최댓값 사이에서) 랜덤한 임곗값을 골라 데이터셋을 둘로 나눈다. 이런 식으로 데이터셋은 점차 분리되어 모든 샘플이 다른 샘플과 격리될 때까지 진행된다. 이상치는 일반적으로 다른 샘플과 멀리 떨어져 있으므로 (모든 결정 트리에 걸쳐) 평균적으로 정상 샘플과 적은 단계에서 격리된다.
=>LOF(local outlier factor) : 이상치 탐지에 좋은 알고리즘, 주어진 샘플 주위의 밀도와 이웃 주위의 밀도를 비교한다. 이상치는 종종 k 개의 최근접 이웃보다 더 격리된다.
=>one-class SVM : 특이치 탐지에 좋은 알고리즘, 고차원 데이터 셋에 잘 작동, 모든 SVM 과 마찬가지로 대규모 데이터셋으로의 확장은 어렵다

* 군집을 어떻게 정의할 수 있나요? 몇 개의 군집 알고리즘을 말해보세요.
=> 머신러닝에서 군집은 비슷한 샘플을 모으는 비지도 작업니다. 유사도 개념은 주어진 문제에 따라 다르다. 예를들어 어떤 경우에는 가까이 있는 두 샘플이 비슷하다고 생각할 수 있지만 다른 경우에는 조밀하게 모여 있는 그룹에 같이 속해 있는 한 멀리 떨어진 샘플도 비슷하다고 볼 수 있다. k-평균, DBSCAN, 병합 군집, BIRCH, 평균-이동, 유사도 전파, 스펙트럼 군집 등이 인기가 많은 군집 알고리즘이다

* 군집 알고리즘의 주요 애플리케이션은 무엇인가요?
=> 군집 알고리즘의 주요 애플리케이션은 데이터 분석, 고객 분류, 추천 시스템, 검색 엔진, 이미지 분할, 준지도 학습, 차원 축소, 이상치 탐지, 특이치 탐지 등입니다.

* k-평균을 사용할 때 적절한 클러스터 개수를 선택할 수 있는 두 가지 기법을 설명하세요.
=> k-평균을 사용할 때 클러스터 개수를 선택하는 간단한 방법은 엘보 규칙이다. 클러스터 개수의 함수로 이너셔(각 샘플과 인접한 센트로이드 사이의 평균 제곱거리)를 그리고 그래프에서 이너셔가 더는 빠르게 감소하지 않는 지점(엘보)을 찾는다. 일반적으로 이 지점이 최적의 클러스터 개수에 가깝다. 다른 방법으로는 클러스터 개수의함수로 실루엣 점수를 그래프로 그린다. 그래프에 뾰족하게 올라간 지점이 나타나는 경우가 많은데 일반적으로 이 근처가 최적의 클러스터 개수이다. 실루엣 점수는 모든 샘플에 대한 평균 실루엣 계수이다. 샘플이 자신의 클러스터 안에 있고 다른 클러스터와 멀리 떨어져 있다면 +1 에 가깝고 다른 클러스터에 매우 인접해 있으면 -1 에 가까워진다. 실루엣 다어이그램을 그려 좀 더 많은 분석을 수행할 수 있다.

* 레이블 전파는 무엇인가요? 왜 이를 구현해야 하고 어떻게 구현할 수 있나요?
=> 데이터셋에 레이블을 부여하는 것은 비용과 시간이 많이 든다. 따라서 보통 레이블이 없는 샘플은 많고 레이블이 있는 샘플은 적다. 레이블 전파는 레이블이 있는 샘플의 일부(또는 전부)를 레이블이 없는 비슷한 샘플에 복사하는 기법이다. 레이블을 가진 샘플의 개수를 크게 늘릴 수 있고 지도 학습 방법을 사용해 더 나은 성능을 낼 수 있다.(이것이 준지도 학습 방법이다) 이를 구현하는 한 가지 방법은 k-평균과 같은 군집 알고리즘을 모든 샘플에 적용한 다음 군집마다 가장 많은 레이블이나 가장 대표적인 샘플(센트로이드에 가장 가까운 샘플)을 찾아 동일 클러스터 안에 있는 레이블이 없는 샘플에 전파하는 것이다.

* 대규모 데이터셋으로 확장할 수 있는 군집 알고리즘 두 개를 말해보세요. 밀도가 높은 지역을 찾는 군집 알고리즘 두 개는 무엇인가요?
=> k-평균과 BIRCH 를 대규모 데이터셋에 적용할 수 있다. DBSCAN 과 평균-이동이 밀도가 높은 지역을 찾는다.

* 능동 학습이 유용한 경우는 언제 인가요? 어떻게 구현할 수 있나요?
=> 레이블이 없는 샘플이 많고 레이블을 부여하는 것에 비용이 많이 들 때는 능동학습이 유용하다. 이런 (매우 흔한) 경우에는 무작위로 샘플을 선택해 레이블을 부여하는 것보다 능동학습이 더 바람직하다. 전문가가 학습 알고리즘과 상호작용하여 알고리즘이 요청하는 특정 샘플에 레이블을 제공한다. 널리 사용되는 것은 불확실성 샘플링이다.

* 이상치 탐지와 정상치 탐지의 차이는 무엇인가요?
=> 이상치 탐지는 이상치가 포함될 수 있는 데이터셋에서 알고리즘을 훈련한다. 이상치 탐지의 목표는 전형적으로(훈련 세트 안에 있는)이상치와 새로운 샘플 사이에 있는 이상치를 구별해내는 것이다.
특이치 탐지에서는 '깨끗'하다고 가정한 데이터셋에서 알고리즘을 훈련한다. 이 알고리즘의 목적은 새로운 샘플 사이에서 특이한 것을 감지하는 것이다. 어떤 알고리즘(예를 들어 아이솔레이션 포레스트)은 이상치 탐지에 최적인 반면 다른 알고리즘(예를 들어 one-class SVM)은 특이치 탐지에 잘 맞다

* 가우시안 혼합이 무엇인가요? 어떤 작업에 사용할 수 있나요?
=> 가우시안 혼합 모델(GMM)은 샘플이 파라미터를 모르는 몇 개의 가우시안 분포에서 생성되었다고 가정하는 확률 모델이다. 다른 말로 하면 데이터가 유한한 개수의 타원 모양 클러스터로 그룹 지어 있다는 가정이다.(클러스터의 타원 모양, 크기, 방향, 밀집도는 다를 수 있다.) 하지만 샘플이 어떤 클러스터에 속해 있는지는 알지 못한다. 밀집도 추정, 군집, 이상치 탐지에 이 모델을 사용할 수 있다.

* 가우시안 혼합 모델을 사용할 때 적절한 클러스터 개수를 찾는 두 가지 기법을 말해보세요.
=> 가우시안 혼합모델을 사용할 때 적절한 클러스터 개수를 찾는 한가지 방법은 클러스터의 개수의 함수로 BIC 나 AIC 그래프를 그리는 것이다. 그 다음 BIC 나 AIC 를 최소화하는 클러스터 개수를 선택한다. 또 다른 방법은 베이즈 가우시안 혼합 모델을 사용해 클러스터 개수를 자동으로 선택하는 것이다.