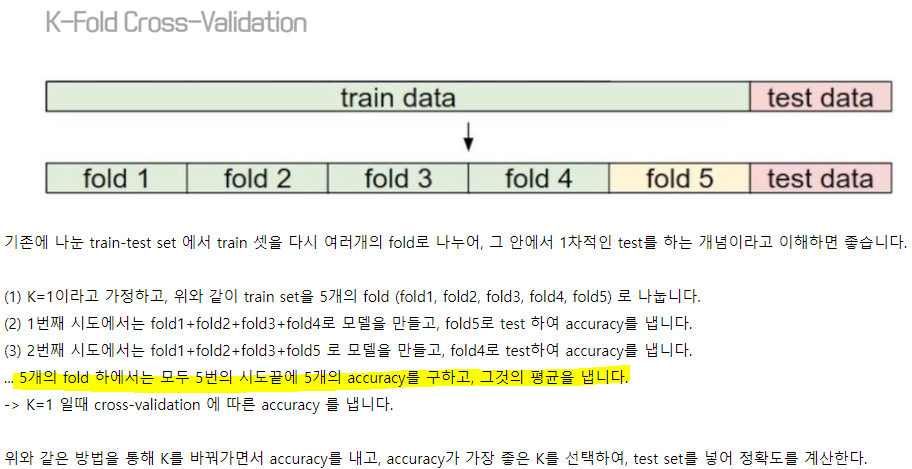
머신러닝 알고리즘 장단점 비교

**1. KNN(최근접이웃)**

-> knn알고리즘은 최근접 이웃 알고리즘으로써 데이터간의 거리를 기준으로 거리가 가장 가까운 k개의 데이터를 기준 삼아서 새로운 데이터를 분류하는 방법이다. 이때 데이터의 적합성과 정확도를 높이기 위해서 적절한 k를 선정하는 것이 굉장히 중요하다.

예를 들어 k가 굉장히 작게 된다면 분류기준이 굉장히 제한 되기 때문에 과적합이 일어난다. 또한 k가 굉장히 크게 된다면 너무 많은 기준으로 데이터를 분류하기 때문에 새로운 데이터에 대해서 모델이 정확하게 판단하지 못하므로 언더피팅이 일어난다.

적절한 k값을 찾기 위해서 교차검증방법을 사용한다. Cross validation은 과적합을 피하기 위해서 주로 사용하는 방식으로 k-fold cross validation이 있다.

**<cv 한번에 이해>**

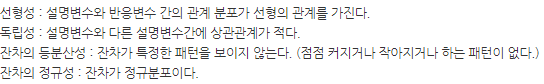
Knn은 연속형 자료를 예측하는 문제에서 사용되지 않고 범주형 자료를 거리에 기반해서 분류하는 알고리즘이다

* 장점 : 이해하기 쉽고 조절할 파라미터가 data distance(맨하탄, 유클리디안), K 인데 많이 조정하지 않아도 좋은 성능
* 단점 : 훈련세트가 크게 되면 예측이 느려짐, 예측이 대체적으로 느리고 특성 처리가 부족 과적합, 과소적합 자주 발생
* Knn은 분류문제에 적합하며 회귀문제에는 사용x

**2. 선형회귀모델**

-> 회귀와 분류에서 모두 사용하는 모델로써 주어진 데이터를 대표하는 하나의 직선을 찾는 것이다. 이 직선을 회귀선이라 부르며 이 선을 함수로 표현한 것을 회귀식이라고 한다. 해당 모델은 최소제곱법(최소자승법)을 기준으로 회귀선을 찾는다. 즉, 최소제곱법은 잔차의 제곱의 합이 최소가 되도록 하는 직선을 회귀선으로 하는 것을 의미하며 회귀식을 바꿔가면서 잔차의 제곱 합이 최소가 되게 하는 직선을 찾는 것이 **선형회귀분석**

* 장점 : 학습 속도가 빠르고 예측도 빠르다. 매우 크거나 작은 데이터에서도 사용가능, 예측이 어떻게 이뤄지는지에 대한 수식이 비교적 쉬움. 선형 모델은 샘플에 비해서 특성 즉, 변수가 많을 때 잘 작동한다.
* 단점 : 선형성을 만족하게 예측하는 회귀식이므로 즉, 회귀식에 모이도록 데이터 값을 예측함 그래서 예측하는 데이터 자체가 주기성을 갖게 된다. 그리고 데이터셋의 특성들끼리 서로 깊게 연관 되어 있을 때는 계수와 예측이 불안정하다. 저차원의 데이터셋에서는 적합하지 않다.

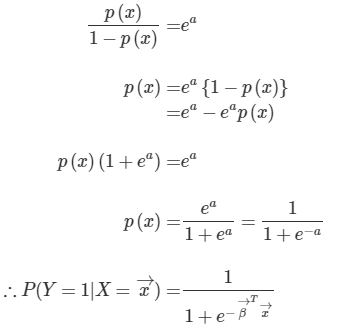


**2-1. 로지스틱 회귀분석**

-> 숫자 그 자체로 의미를 가지는 연속형 변수를 예측하거나 검정하는데 사용되는 선형회귀모델로는 범주형 자료를 분류 예측할 수 없기 때문에 생긴 분석이다. **시그모이드 함수**를 사용하며 로지스틱 함수를 항상 출력 값이 0~1사이의 값이 된다. 즉, **확률밀도함수(pdf)** 요건을 충족하는 함수이다.

로지스틱 회귀분석에서 중요한 것은임의의 사건 A가 발생하지 않을 확률 대비 일어날 확률의 비율인 odds라는 식이다.

이때 해당 식에 log를 취하고 정리하면

가 도출된다.

즉, 로지스틱회귀분석은 정확히 0,1로 분류하는 것이 아니라 0일 확률, 1일 확률을 구하는 것이다.

또한 로지스틱 회귀분석에서

의 x는 -inf ~ inf 의 값을 갖고 있으나 y는 0~1값을 갖는 확률이기 때문에 y도 -inf~inf값을 갖도록 해야 한다. Y를 확률값 p로 바꿔보면

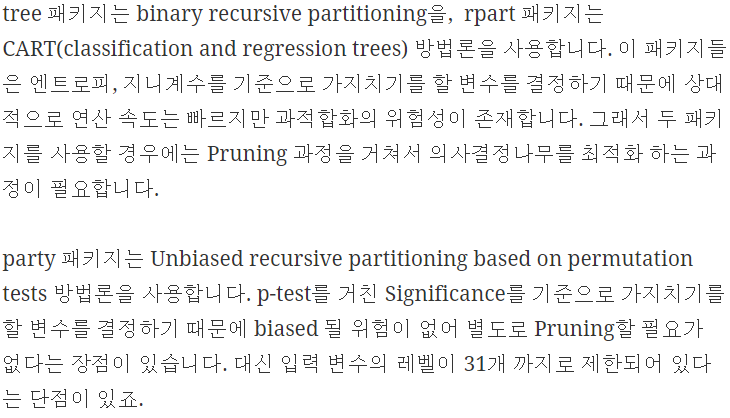
이 되며 p는 위의 식처럼 odds로 나타낼 수 있다. 하지만 odds도 마찬가지로 0~1사이의 값을 갖고 잇는데 이를 -inf~inf로 바꿔주기 위해서 odds에 log를 취한 값을 p로 받는다면 양변의 수치 범위가 같아진다. 그 이후에는 위의 식대로 진행하면 된다.

🡪로지스틱 회귀분석은 선형회귀에서 못하는 분류작업을 위해서 진행하는 분석으로 수치를 예측하는 부분에서는 맞지 않다.

**3. 나이브베이지안분류**

**4. Decision tree**

-> 의사결정 나무는 머신러닝중 하나로 특정 항목에 대한 의사 결정 규칙을 나무 형태로 분류해 나가는 분석기법을 말한다. 분석과정이 직관적이고 시각화하기가 편리하고 이해하기가 쉽다. 인공신경망처럼 블랙박스모델이 아닌 분석과정에 따른 결과를 눈으로 관찰할 수 있는 화이트박스 모델이다. 그리고 연속형/범주형 변수 모두를 사용할 수 있으며 대규모 데이터 분석에서도 빠르게 연산이 가능한 것이다. 의사결정나무는 CART와 CHAID알고리즘 C5.0등의 알고리즘이 존재한다. 의사결정나무는 tree, rpart, party의 패키지가 존재하며 해당 패키지들에 대한 설명은 아래에 있다.

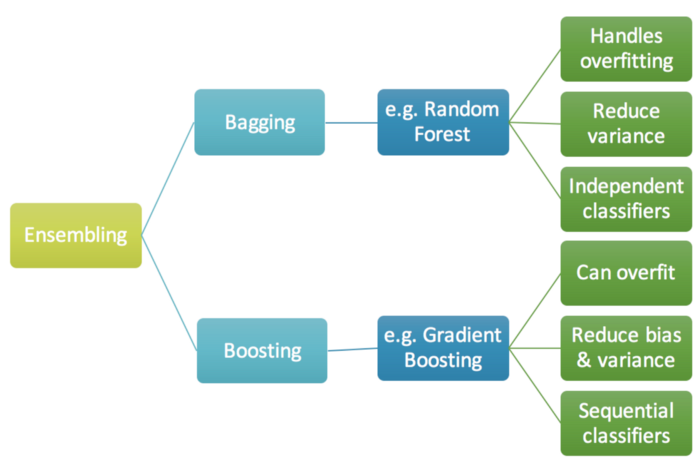


🡪 decision tree방법은 단순하고 훈련 데이터 밖의 새로운 데이터를 예측하는 능력이 없기 때문에 사용x 그리고 시계열 데이터에 적합하지 않으며 트리의 깊이가 깊어질수록 매우 장황해지며 pruning과정을 거쳐도 과적합 될 가능성이 있어 일반화 성능이 좋지 않다.

따라서 앙상블의 대표적인 기법들 bagging, boosting방법을 거쳐서 회귀 문제의 다양한 데이터 셋에 대응할 수 있는 모델을 구성해야 한다.

**5. Ensemble**

->의사결정 나무의 집합체 즉, 숲이라고 보면 되며 대표적으로 Bagging방법과 Boosting방법이 있다.



**Bagging**

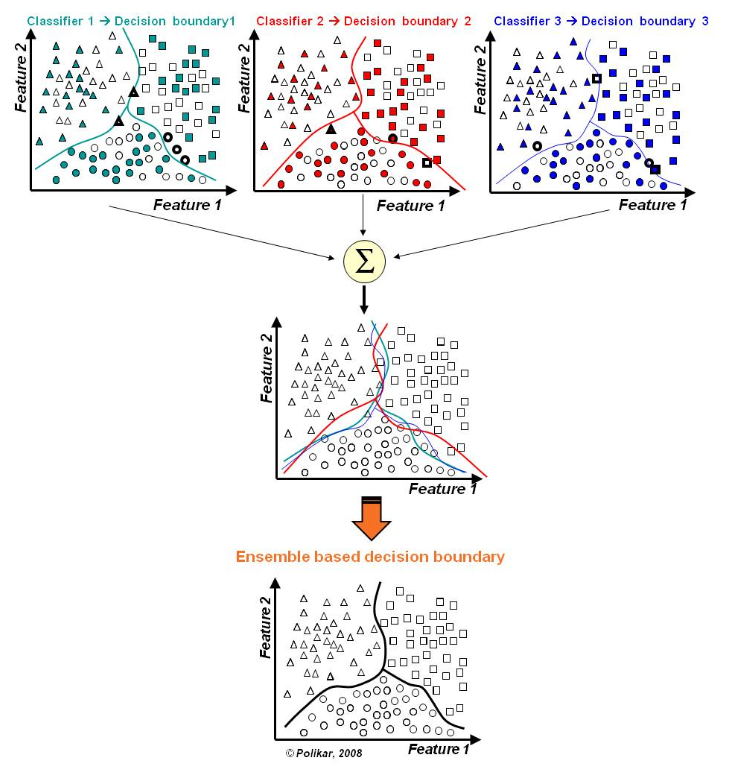
**🡪**bagging방법은 샘플을 여러 번 뽑아 각 모델을 학습시켜 결과를 집계 하는 방법이다. 대상 데이터로부터 복원 랜덤 샘플링을 진행한 뒤 추출한 데이터가 일종의 표본 집단이 되어 각 표본 데이터에 동일한 모델을 학습시킨 뒤 학습된 모델에 대한 예측 변수들을 집계하여 그 결과로 모델을 생성한다. 이를 **Bootstrap Aggregating**라고 부른다. – 병렬처리를 따름

이렇게 bagging방법을 사용하여 새로운 알고리즘을 만드는 이유는 알고리즘의 안정성과 정확성을 향상시키기 위해서이다. 보통 모델에서 나타날 수 있는 오류는 다음과 같다.

1. 높은 편향으로 인한 언더피팅

2. 높은 분산으로 인한 오버피팅

앙상블 기법의 배깅은 각 샘플에서 나온 결과를 중간값으로 맞춰주기 때문에 오버피팅을 피할 수 있다.(categorical data = 투표방식으로 집계 / continuous data = 평균으로 집계)

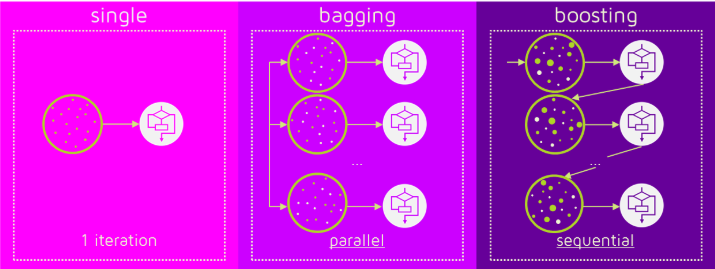


-> 대표적인 bagging algo로는 RF가 있는데, 단일 decision tree는 기준선이 굉장히 discrete한 모양이지만 bagging방식을 거쳐서 집계된 앙상블 모델은 discrete한 모양을 벗어날 수 있다.

**Boosting**

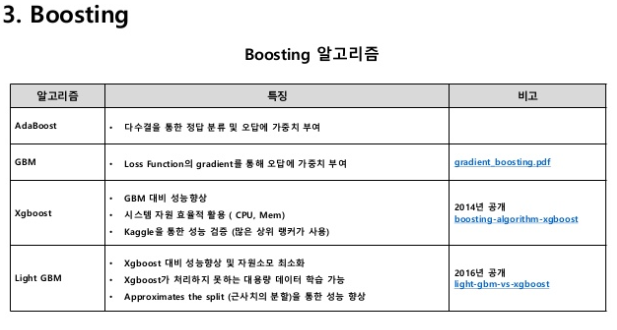
-> 부스팅 방법은 약한 신호기를 가진 분류기들을 합쳐서 강한 신호를 가진 분류기로 만드는 방법을 말한다. Boosting은 bagging과 동일하게 복원샘플링을 하지만 가중치를 부여하는 면에서 차이가 있다. Boosting는 학습을 sequential하게 시키며 학습이 끝나면 나온 결과에 기반해서 가중치를 매긴다.

즉, 부스팅 방식은 A,B,C 분류기의 각 성능이 0.3일 때 A,B,C의 분류기를 결합하여 0.7의 정확도까지 올리는 방식이다. 따라서 A분류기를 만들고 그 정보를 바탕으로 B, 그리고 C이렇게 반들어 최종적으로 만들어진 분류기들을 모두 결합하여 최종 모델을 만드는 방식이다.



오답에 대해 높은 가중치를 부여하고 정답에 대해 낮은 가중치를 부여하기 때문에 오답에 더욱 집중할 수 있게 된다. 그래서 정확도를 높일 수 있지만 그만큼 이상치에 취약하다.

대표적인 종류로는 Adaboost, xgboost, gradientboost가 있다.



**6. Ensemble-RF**

-> random foreest는 가장 대표적인 bagging의 방법 중 하나로써 과적합과 과소적합을 피하면서 데이터를 학습시켜 예측/분류하는 알고리즘을 말한다.

Rf는 매개변수 튜닝이 없이도 단일 decision tree보다 높은 정확도를 낸다. 그리고 하이퍼 파라미터 조정 없이 defaultt값으로만 좋은 결과를 만들어 줄 때가 많다. 또한 데이터의 스케일(표준화)을 맞출 필요가 없어 사람들이 많이 이용하는 알고리즘 중 하나이다

하지만 비정형 데이터 같이 매우 차원이 높은 희소한 데이터에는 잘 작동하지 않는다. Rf는 학습 시키는 시간이 굉장히 오래걸리며, 더 많은 트리를 평균낸다면(continuous 변수) 과대적합을 줄여 안정적인 모델을 만들 수(무작위 샘플을 복원추출한 표본들에 대해 각각 서로 다른 방향으로 과대적합된 단일 트리 모델을 학습시킨다 -> 과적합이 일어날 수 있다 -> 따라서 방지하기 위해서 평균을 산출) 있지만 시간이 굉장히 오래 걸린다.

**7. Ensemble-gradient boosting regression tree(GBM)**

<https://3months.tistory.com/368> -> 설명, 수식 위주

<http://uc-r.github.io/gbm_regression> -> 코드 위주

GBM은 residual fitting로 이해하는 것이 가장 쉽다. 모델 A로 예측하고 예측하지 못한 잔차들을 다시 모델 B로 예측하고 B의 잔차를 C로 예측하는 이러한 과정을 통해서 잔차를 줄일 수 있고 훈련데이터를 잘 설명하는 장점이 있다. 하지만 이는 과적합의 문제도 발생한다.

쉽게 말하자면 boosting기반 알고리즘은 잔차를 계속 줄여나가는 방식으로 분류기들을 연결지어서 학습시켜 강한 분류기를 만드는 것인데 보통 약한 분류기로 decision tree를 사용한다.

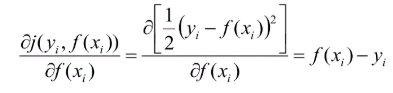
**Residual fitting와 gradient의 관계 설명**

Residual은 loss funcition을 squared error로 설정했을 때 negative gradient이다. 다라서 residual fitting해서 다음 모델을 순차적으로 만들어 나가는 것이 negative gradient를 이용하여 다음 모델을 순차적으로 만들어 나가는 것이다.

Loss function



Negative gradient = residual



Residual



따라서 gradient boosting는 다음 모델을 만들 때, negative gradient를 이용하여 만들기 때문에 gradient boosting라고 할 수 있다. 즉, 직관적으로 어떤 데이터 포인트에서 loss function이 줄어들기 위해 함수 f(x)가 가려고 하는 방향을 말하며 이 방향에 새로운 모델을 fitting해서 이전 모델과 결합하면, f(x) 는 loss function이 줄어드는 방향으로 업데이트가 된다.

* 장점 : gradient boosting tree는 보통 1~5의 깊지 않은 트리를 사용하므로 메모리를 적게 사용해서 예측 속도가 빠르다. 그리고 rf보다는 매개변수 설정에 더 민감하지만 최적의 하이퍼파라미터를 찾으면 rf보다 더 높은 정확도를 제공한다. 모든 지도학습에서 가장 많이 사용되는 모델이다. 변수별로 스케일을 조정하지 않아도 되고 반응성이 가장 좋다.
* 단점 : rf보다는 매개변수 설정에 더 민감하며 따라서 매개변수 설정을 잘 조정해야 한다. 다라서 random search보다 grid search방식을 활용하여 적합 변수를 찾아야한다. 따라서 grid search를 사용했을 시 시간이 오래걸린다.

🡪 해당 모델은 예측 분류 모두 사용 가능하며 rf비교하여 이전 트리의 오차를 개선해가는 방식으로 순차적으로(sequential) 트리를 만든다. 그리고 무작위성이 없으며 강력한 사전 가지치기가 사용된다. 약한 분류기들을 합치는 것으로 트리가 많이 추가될수록 성능이 좋아진다. 시간적인 부분에서는 rf보다 gbm을 사용

**8.** **Xgboost**

-> 방식은 gbm과 같은 boosting방식이며 대부분 gbm과 특성이 비슷하다고 생각하면 된다. gbm다음으로 나온 알고리즘으로 성능이 더 좋다 특히 장점으로는 훌륭한 gradient boosting library이며 병렬 처리를 사용하므로 학습과 분류가 빠르다. 그리고 유연성이 좋아 다양한 커스텀에 최적화 옵션을 제공한다. 그리고 자동 가지치기를 하므로 과적합이 잘 일어나지 않으며 다른 알고리즘과 연계활용성이 좋다. (앙상블)

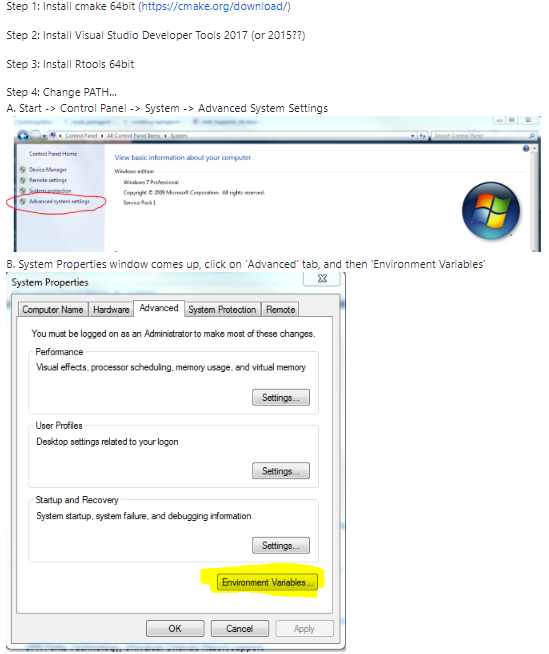
**9. Adaboost**

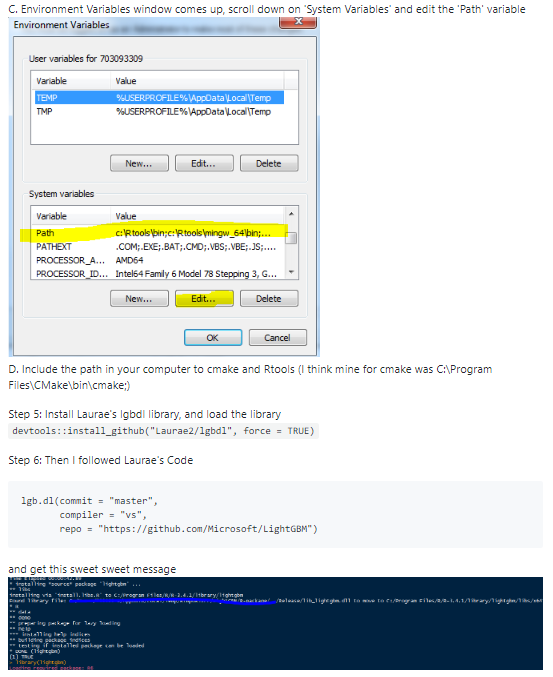
-> 가장 초창기의 부스팅 모델로써 가중치를 통한 분류하는 문제에서 사용 즉, 모델에서 잘못 예측한 데이터 오차 데이터에 가중치를 부여하고 잘못 분류한 데이터를 분류하는데 더 집중을 하고 다음 모델은 또한 모델1,2에서 잘못 예측한 데이터를 분류하는데 집중한다.

비용함수(cost function)인 가중치를 반영하여 계산

**10. Light GBM**

**\*설치**



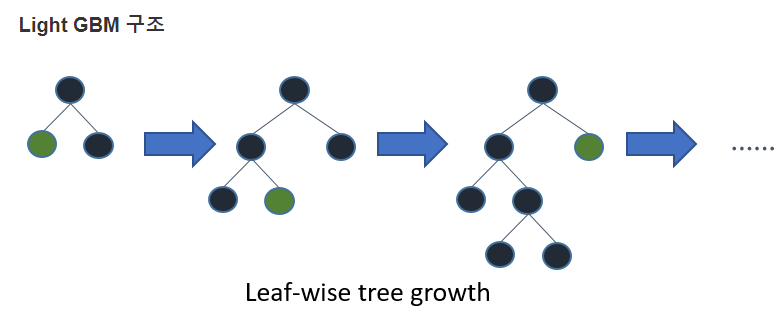


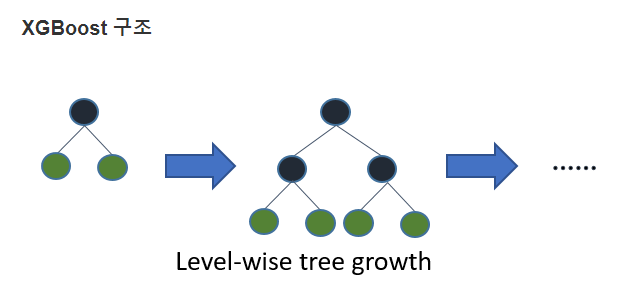
<https://www.slideshare.net/freepsw/boostin,g-bagging-vs-boosting> -> 앙상블 알고리즘 자세한 정리

light gbm code in r <https://rpubs.com/awanindra01/lightgbm> .

<https://rpubs.com/awanindra01/lightgbm>

-> xgboost대비 성능향상 및 자원소모 최소화, 대용량 데이터 학습가능, 근사치 분할을 통한 성능 향상, 오버피팅을 줄일 수 있다. Gbm은 가지가 1번 분리될 때 2개씩 매번 분리가 되어 과적합이 잘 일어나지만 lightgbm은 가지가 1번 분리 될 때 모든 가지를 분리하지 않고 2개의 node중 잘 맞는 기준으로만 분리한다 아래는 architecture





Xgboost와 gbm은 둘 다 위와 같은 구조로 되어 있어 계속 둘중에 한 개가 아닌 두 node가 가지로 분리되므로 굉장히 복잡한 모델이 형성되어 과적합을 야기하기가 쉽다 하지만 lightgbm은 과적합이 일어날 확률이 더 적어지는 것이다. 하지만 모델링을 할 때 어느 모델이 나은지는 직접 부딪혀봐야 아는 법! 정확도와 logloss값을 확인해봐야한다.

대량의 데이터를 병렬로 빠르게 학습가능한 장점 메모리 소모가 적다. – 대량의 데이터 학습에 적합

예측 정확도가 더 높다 과적합에 민감

동일한 hyper parameter를 설정시 xgboost보다 2~10배 정도 빠르다. Lightgbm은 2017년도부터 시행됐기 때문에 최적화가 잘 안되어있음

31
3-4. Light GBM
Light GBMìì ë°ì´í°ë¥¼ ì ë¶í íë ìì±ì ë³ë ¬ë¡ ì°¾ë ë°©ë²
ì´ë»ê² Best Split(ìµì  ë¶í ê°)ì ì°¾ëê°? Efficient Parallel Algorithm for Deci...

But 오픈소스이긴 하지만 설치가 굉장히 까다로움 아직 설치 못함…

스태킹

8. 커널기법

**9. SVM(커널기법 기반**)

-> 직선과 초평면은 유연하지가 않아 저차원 데이터셋에서는 선형 모델에 매우 제한적이다. 이때 선형모델을 유연하게 만드는 방법은 특성끼리 곱하거나 거듭제곱하는 식으로 새로운 특성을 추가하는 것이다(교호항을 만드는 것) 데이터셋에 비선형 특성을 추가하면 선형모델을 강력하게 만들 수 있다. 하지만 변수가 많은 경우 어떤 특성을 추가할지 모르며 특성을 많이 추가하게 되면 연산 비용이 굉장히 커진다. 다행이게도 커널 기법을 이용하면 실제로 데이터를 확장하지 않고 확장된 특성에 대한 데이터 포인트들의 거리(스칼라의 곱)를 계산한다.

SVM은 데이터의 특성이 몇 개 안되더라도 작동하고 저차원, 고차원 데이터에서 모두 잘 작동한다.

하지만 샘플이 많은 즉, 데이터가 10000개가 넘어가버리면 속도와 메모리 관점에서 좋지 않다. 그리고 데이터 전처리와 매개변수 설정에 신경을 많이 써서 BAGGING, BOOSTING기반의 알고리즘을 이용한다. 그리고 SVM모델은 분석하기도 어려워 예측이 어떻게 진행되었는지 알지 못하고 모델설명이 난해하다. 하지만 모든 특성이 비슷한 단위이고 스케일이 비슷하면 SVM을 시도할 만함 그래서 암호화폐가격 데이터에 SVM이 유용

#데이터셋의 단위와 스케일이 비슷한 형태에서 svm이용

#svm은 로지스틱회귀나 분류rf 등 두개 이상으로 나누어진 집단을 분류하는데 사용하기도 하며 연속형 변수를 예측하는 회귀에서도 사용한다. (단, 이름이 svr임)

#svm은 분류 알고리즘에서 정확도가 높은 편이며 이상치의 영향도 적게 받음

#기본 방법은 데이터를 나누는 "최적의 경계"를 만드는 방식임

#경계선에서 가장 가까운 벡터를 support vecto라하고 데이터와 경계 사이의 거리를 마진이라고 하고 이 margin에서 가장 가까운 데이터를 support vecto라함

#svm에서는 커널 트릭이라는 기법으로 초평면을 3차원으로 그릴 수 있음

#이 기법을 이용하여 2차원으로 분리하기 어려운 데이터도 분리

#but svm은 차원이 많고 데이터수가 많으면 굉장히 오래 걸리는 단점이 있음

#분류예측 모델링

11. 판별분석

12. 릿지 – 분류확률

13. 라쏘 – 분류확률

14. 엘라스틱넷 – 분류확률

* **따라서 rf + gbm + lightgbm + xgboost 사용 SVM은 신체치수데이터의 단위가 모두 다르기 때문에 사용 X**

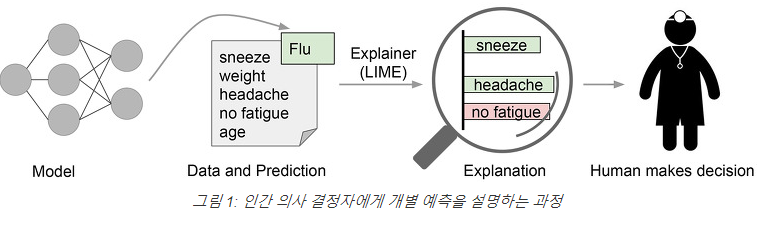
**#라임알고리즘(사후 모델)**

참고자료 - <https://nhlmary3.tistory.com/entry/LIME-Locallly-Interpretable-Modelagnostic-Explanation>

<https://dreamgonfly.github.io/2017/11/05/LIME.html> - 파이썬 코드, 정리 대박!

딥러닝에 사용되는 Deep Neural Networks, RF, Boosting, Ensemble 등은 모델링을 한 후 모델의 해석력을 통해서 인사이트를 도출하는데 돕는 역할을 하며 의사결정나무와는 다르게 어떤 변수가 어떻게 들어가서 얼마나 영향을 끼치는지 모르는 블랙박스 모델이다.

이에 Lime알고리즘은 모든 예측 모델에 대한 결과를 해석 가능하고 신뢰할 수 있는 방법으로 설명하는 새로운 기법을 제공하는 알고리즘으로, 설명하고 싶은 예측 값 근처에 대해서만 해석 가능한 모델을 학습시키는 방법이다.



<특정환자가 독감에 걸렸는지 걸리지 않았는지 여부를 분류 예측하는 모델 도표>

위에 그림과 같이 예측결과는 설명자에 의해 설명되며 설명자는 모델에서 가장 중요한 몇가지 증상(변수)을 강조하여 알려준다. 따라서 의사가 독감에 걸리는 이유에 대해서 구체적으로 알아낸다면 의사는 더 나은 결정을 내릴 수 있을 것이다.

**글로벌 해석력** – 학습된 반응 함수에 의해 모형화 된 조건분포 전체를 이해하는데 도움을 준다. 하지만 해석이 대략적이거나 평균값에 근사함

**로컬 해석력** – 단일 데이터 포인트나 분포의 작은 영역에 대한 이해를 돕는다. 그것은 한 그룹(데이터)의 레코드, 즉 단일 데이터에 대한 값을 도출해준다. 그리고 조건분포의 작은 영역이 선형일 가능성이 높아 글로벌 해석력보다 설명력이 더 높을 것이다. 즉, 라임은 글로벌 해석력보다 로컬 해석력을 제공하도록 설계되어 있어 특정 결정이나 결과에 정확하다.

**라임의 장점**

라임은 로컬 설명을 학습하기 위해 분류기의 결정 경계(decision boundary)를 해석가능한 모델을 이용하여 근사한다. 라임은 모델에 영향을 받지 않는 독립적인 모델이기 때문에 모델 행동에 대한 어떠한 가정, 제약을 가하지 않는다. 따라서 라임은 어떤 예측 모델이든지 적용할 수 있다.

라임이 블랙박스 모델을 설명하는 방법은 해석 가능한 간단한 선형 모델로 근사하는 것이다.

#텍스트분석에서의 사용 알고리즘

감정분석

W2V

clustring