**Classification 머신러닝 참고자료**

머신러닝자료: <https://github.com/Youngpyoryu/SD_academy/blob/732f017936ae6e7c8295ffa7dbbcdf4fe21302c1/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D/%EB%B6%84%EB%A5%98(Classification)/Classification_210413.ipynb>

2021/04/13

KNN

특정구간에서는 1, 특정구간에서는 0 이런식으로 특정 구간 별로 값이 나뉨.

통계학파 크게 두 가지로 나뉨. 1. 빈도주의(실제로 일어난 일) 2. Bayesian

P(A|B) = P(B|A)P(A)/P(B)

여기서 P(B|A)는 사전확률

P(A)/P(B)는 Likelihood

P(A|B)는 사후 확률을 의미한다.

Ex) P(비|맑은날) -> 맑은 날 중에서 비가 올 확률

Entropy 가 높다 라는 것은 화합물(혼잡도)이 많다는 것.

Entropy 가 낮다 라는 것은 화합물(혼잡도)이 적다는 것

-log 함수는 부호가 (-)이므로 정보량이 증가할수록 확률이 감소함.

log함수가 x축을 기준으로 대칭 되는게 -log함수

지니계수가 낮으면 데이터가 잘 분리 되었다.

지니계수가 높으면 데이터가 잘 분리 되지 않았다라는 의미

2021/04/14

**Decisoin Tree**

DecisionTree Classifier.html

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

1/P(X) -> -log(P(X) 로 치환을 통해 계산하기 쉽게 바꿈

단조증가함수, 역함수때문에 치환함. log를 취함으로써 값이 바뀌긴 하지만, 그래도 log를 취해야 계산하기 쉽다.

Entropy p.11 PPT 크로스 엔트로피 손실함수

log의 밑은 2 : 이진분류때문에 2로 한다. 010101010010110010101

X= 사건 (다 모아놓은 것)

x = 어떤 특정 사건

E(-logP(x)] = 시그마 -logP(x) P(X)

특정 사건에 취하면

정보량 = 1/P(X) (확률분에 일)

확률이 적다는 것은, 정보량이 많다는 것

X=x

특정 사건을 우변에 넣기 때문에 x에는 들어가나, P(X)는 그대로

x -> -logP(x)로 치환

X -> 정보량

사건이 정보량으로 변하면서 log가 안으로

p.11 2번째 줄 E[-logP(x)] -> E[-logP(X)]로 수정

로그의 성질

logaX =y

x>0, a>0 & a!=1

information gain rate 부모노드와 자식노드 간의 비교를 나타낸다.

gini index는 자식 노드 안을 보여준다.

entropy, gini가 낮아지게끔

IG, GR은 확인만 함.

최대한 가지가 적을 수록 좋음

Decision Tree 추가 설명

<https://leedakyeong.tistory.com/entry/Decision-Tree%EB%9E%80-ID3-%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98>

<https://wooono.tistory.com/104>

시각화 도구

graphviz 설치

<https://graphviz.org/download/#windows>

vscode 꾸미기

<https://rural-mouse.tistory.com/15>

<https://marketplace.visualstudio.com/items?itemName=vscode-icons-team.vscode-icons>

np.where(cond.xarr.yarr)

ex)

cond=[False, True, False, True]

xarr=[1,2,3,4]

yarr=[100,200,300,400]

np.where(cond.xarr.yarr)

cond-> true -> xarr

cond-> false -> yarr



mglearn은 가상의 그림도 그릴 수 있음.

2021/04/15

부트스트랩에서

다수결의 원칙 = 평균 원칙

배깅은 훈련세트에서 중복을 허용하여 샘플링하는 거고

오버샘플링은 데이터를 극단적으로 늘리는 것

부스팅은 배깅과 달리 Decision Tree끼리 독립이 아님

부스팅 中 Ada Boosting



Gradient Boosting



-> 잔차를 줄여간다.

encoding 하는 방법 종류

<https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02>

손실함수

<http://www.gisdeveloper.co.kr/?p=7631>



XGBoosting은 칼럼으로 접근하기 때문에 NaN값이 나오면 그냥 건너 뛰는게 가능하다.

xgboost 공식문서

<https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/>

Nvida tool kit(CUDA)

<https://developer.nvidia.com/cuda-toolkit>

cuda를 깔아야 돌아감

VScode 테마: Tinacious Design Theme