Traditional Programing Data, Program → ouput

Machine learning Data, output → Program

Supervised learning → training data labeled (train on output)

Unsupervised learning → train data unlabeled need to find hidden structure

semisupervised learning → only part of input is labeled

reinforcement learning → Learner interacts to find optimal behavior

Deep learning → Automatic learning without human interference

Regression → Predict a number

Classification → Predict categories

- Clustering (group data)
- Dimensionality Reduction (compress data)
- Anomaly detection (find outliers)

Unsupervised learning Application

when to split? max purity

limit depth of trees to → doesn't get to big/widely, less prone to overfit

stop splitting when node 100% one class, when purity improvement below threshold, when number of examples below threshold

Entropy (impurity)

when =1 most impure, =0 most pure, p_1 = fraction of ex. are cats?

$H(p_1)$ = entropy, $p_0 = (1-p_1)$ fraction not in ex. not cats

Entropy with multiple → $H(p_1) = -\sum_i p_i \cdot \log_2(p_i)$

$H(p_1) = -p_1 \log_2(p_1) - p_0 \log_2(p_0) = -p_1 \log_2(p_1) - (1-p_1)\log_2(1-p_1)$

Information gain

$H(p_1) - (p_0 H(p_1) + p_0(H_i))$

total - Weighted

One hot encoding for k categories can have k values, 0 or 1

use in Reg tree for Male

ID3 Algorithm → start root calc entropy and IG for all start highest IG

Regression Tree → $H_{parent} - (w_1 H(p_1) + w_1 H(p_0))$

Variance $s^2 = \frac{\sum (x_i - \bar{x})^2}{n-1}$

Decision tree pros → easy understand, little to no data, problematic for error bias, more flexible

Decision trees cons → prone to overfit, high variance, costly

DT sensitive to small data changes, building multiple trees solves this issue

Sampling Replacement → take random but replace before each grab

Bagged DT → given dataset of 21 to B → B between 64 → 128

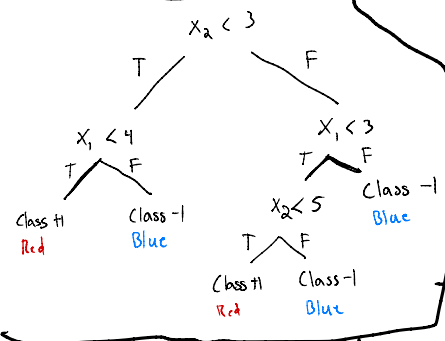
Random Forest Algo → at subset node pick random subset of k<n if n features available, the algo picks max IG typically k = √n

Boosted Trees

Same as bagged DT but must tends to pick examples previously failed

XG Boost → open source of boosted trees, fast efficient, built in step to overfitting

DT + Tree Ensembles → works on structured data, fast, not for unstruct data, works well with multiple models, works slower than DT



$H(Y|C=0) = -(\frac{1}{3}\log_2\frac{1}{3}) - (\frac{2}{3}\log_2\frac{2}{3})$

$= -(.33 \times -1.585) - (.667 \times -.585)$

$H(Y|C=0) = 0.918$

$H(Y|C=1) = -(\frac{2}{3}\log_2\frac{2}{3}) - (\frac{1}{3}\log_2\frac{1}{3})$

$H(Y|C=1) = 0.918$

$IG(C) = H(Y) - (\frac{3}{6} H(Y|C=0) + \frac{3}{6} H(Y|C=1))$

$= 1 - (\frac{3}{6} \times .918 + \frac{3}{6} \times .918)$

$= 0.082$

Gpa	Stud.	Pass
L	F	F
L	T	T
M	F	F
M	T	T
H	F	T
H	T	T

For the left and

$H(Y) = -\sum_{i=1} p_i \log_2(p_i)$

$H(Y) = -(\frac{3}{6}\log_2\frac{3}{6}) - (\frac{3}{6}\log_2\frac{3}{6})$

$= -(.5(-1)) - (.5(1))$

$H(Y) = 1$

$H(Y|A=0) = -(\frac{1}{3}\log_2\frac{1}{3}) - (\frac{2}{3}\log_2\frac{2}{3})$

$= -(.33 \times -1.585) - (.667 \times -.585)$

$H(Y|A=0) = 0.918$

$H(Y|A=1) = -(\frac{2}{3}\log_2\frac{2}{3}) - (\frac{1}{3}\log_2\frac{1}{3})$

$H(Y|A=1) = 0.918$

P pass P repass

$H(Y) = -(\frac{4}{6}\log_2\frac{2}{6}) - (\frac{2}{6}\log_2\frac{2}{6})$

$H(Y) = -(.667 \times -0.585) - (0.33 \times -1.585)$

$H(Y) = 0.917$

$H(Y|\text{Studied} = \text{Yes}) = 0$ All Passed

$H(Y|\text{Studied} = \text{No}) = -(\frac{1}{3}\log_2\frac{1}{3}) - (\frac{2}{3}\log_2\frac{2}{3})$

$= 0.918$

$IG(\text{Studied}) = H(Y) - (\frac{3}{6} H(Y|\text{Studied}=\text{false}) + H(Y|\text{Studied}=\text{T})$

$= 0.917 (\frac{3}{6} \times .918 + \frac{3}{6} \times 0)$

$= 0.458$

A	D	C	Y
0	1	0	Y
1	0	1	Y
0	0	0	N
1	0	1	N
0	1	1	N
1	1	0	Y

For above

For chart above