

SSE - The last term is the sum of squares error, or SSE. The error is the difference between the observed value and the predicted value. We usually want to minimize the error. The smaller the error, the better the estimation power of the regression. Finally, I should add that it is also known as RSS or residual sum of squares. Residual as in: remaining or unexplained. (sum e^2)
The confusion :SST = SSR + SSE ถ้า SSR = SST then we have a perfect fit.

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

Does not tell us if X is the cause of changes in Y

$R^2 = [Cor(Y, X)]^2 = (0.8062)^2 = 0.650$

65% of the total variability in the price is accounted for by the mileage of the car

$t = \frac{\hat{\beta}_1 - 0}{s.e(\hat{\beta}_1)} = \frac{-0.093}{0.00563} = -16.657$

is this significant? Ans: Yes we can accept H₀ with 95% confidence if:


$-t_{\alpha/2, n-2} < t < t_{\alpha/2, n-2}$
 $-1.976 < t < 1.976 \quad (\alpha = 5\%)$

Eager Vs Lazy
Eager - ถูก build ไว้ล่วงหน้า เวลาใช้ unseen data มากี่โยนเข้ามาเดลได้เลย, มีการจูนตลอด เพราะจะไดรับกับ data ใหม่ๆได้
Lazy - Classifier keeps all the training examples for later use.
Data Normalization

$$H(S) = - \sum_{i=1}^n p_i \log_2 p_i$$

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

classification - good features (high purity), bad (low) give less informative
Entropy - In information theory, a measure of uncertainty around a source of information. Low for predictable sources, higher for more random sources.




$p_1 = 6/6 = 1.0$

$p_2 = 0/6 = 0.0$

NB: Define $\log_2(0)=0$


$$H(S) = -((1 \times \log_2(1)) + (0 \times \log_2(0))) = -(0 + 0) = 0$$



$p_1 = 0/6 = 0.0$

$p_2 = 6/6 = 1.0$

$$H(S) = -((0 \times \log_2(0)) + (1 \times \log_2(1))) = -(0 + 0) = 0$$



$p_1 = 3/6 = 0.5$

$p_2 = 3/6 = 0.5$

$$H(S) = -((0.5 \times \log_2(0.5)) + (0.5 \times \log_2(0.5))) = -(-0.5 - 0.5) = 1$$

IG for feature A that splits a set of examples S into {S₁, . . . , S_m} :

$IG(S, A) = (\text{original entropy}) - (\text{entropy after split})$

$$IG(S, A) = H(S) - \sum_{i=1}^m \frac{|S_i|}{|S|} H(S_i)$$

Each subset is weighted in proportion to its size

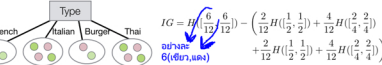
ค.ไม่แน่นอนก่อนตัด - ค.ไม่แน่นอนหลังตัด

original

after split

$$IG = H(\frac{6}{12}, \frac{6}{12}) - (\frac{2}{12} H(\frac{0}{2}, \frac{1}{2}) + \frac{4}{12} H(\frac{1}{2}, \frac{1}{2}) + \frac{6}{12} H(\frac{2}{6}, \frac{4}{6}))$$

$$IG(\text{Patrons}) = 1 - 0.459 = 0.541$$



IG = $H(\frac{6}{12}, \frac{6}{12}) - (\frac{2}{12} H(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} H(\frac{2}{4}, \frac{2}{4}) + \frac{2}{12} H(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} H(\frac{2}{4}, \frac{2}{4}))$

IG(Type) = 1 - 1 = 0

Ensembles
Bagging
Bagging หรือ bootstrap aggregation คือการสุ่มตัวอย่างข้อมูลออกมาแล้วสร้าง classifier ขึ้นมา โดยใช้วิธีสุ่มแบบแทนที่ (random with replacement)
สุ่มข้อมูลหลายๆรอบหรือสุ่ม feature ของข้อมูลเพื่อให้ได้ classifier หลายๆตัว แล้วทำนายโดยใช้ classifiers ทุกตัวที่เราสร้างขึ้นมาเพื่อทำนายชุดข้อมูลใหม่
ถ้าใช้การสุ่มข้อมูลอาจจะเกิดปัญหา OOB out of bag เนื่องจากข้อมูลบางตัวอาจไม่ได้ถูกสุ่มขึ้นมาใช้งานเลย
สามารถช่วยลดการเกิด overfitting ได้เนื่องจากมี classifier หลายๆตัวมาช่วยกันทำนายและหาการใช้การสุ่ม feature ก็ไม่จำเป็นต้องใช้ครบทุก feature ก็ได้
การทำนายก็ไม่ได้หลายแบบได้แก่การเฉลี่ยหรือการโหวตก็ได้ แล้วแต่ว่าเราทำนายความน่าจะเป็นหรือทำนายประเภทข้อมูล
บางคนจะใช้เทคนิคการจำว่า bagging คือการสุ่มข้อมูลมาเป็นหลายๆแล้วสร้างโมเดลจากข้อมูลที่อย่อยออกมาก็ได้
Boosting (เอา classifier ที่ไม่ได้ดีมากมารวมกันจนทำนายข้อมูลที่ซับซ้อนมากๆได้)
Boosting คือการนำ classifier ที่มีความแม่นยำต่ำมา ทำนายข้อมูลที่มี จากนั้นจะให้ weak classifier ตัวใหม่มาแก้ไข error(สุ่มให้เจอบ่อยขึ้น)ที่มี โดยผลรวมของ classifier จะเกิดเป็น classifier ใหม่ขึ้นแล้วจะทำแบบนี้ไปเรื่อยๆจนได้โมเดลที่ดีที่สุดจากผลรวมของ classifier ข้อเสียของการใช้ boosting คือเราต้องรันหลายครั้งและเป็นลำดับกว่าจะได้โมเดลที่ต้องการ
Bagging ช่วยแก้ปัญหา overfitting ส่วน Boosting ช่วยแก้ปัญหาความแม่นยำต่ำ (bias)

C4.5 Algorithm

- C4.5 is an improved version of ID3 algorithm to overcome some of its disadvantages (Quinlan, 1993).
- It contains several improvements to make it "an industrial strength" decision tree learner, including:
 - Handling continuous numeric features.
 - Handling training data with missing values. entropy, gini
 - Choosing an appropriate feature selection measure.
 - Providing an option for pruning trees after creation to reduce likelihood of overfitting. คัด

ที่เอามายกตัวอย่างก่อนหน้านี้

Feature Selection : การเลือก feature มาใส่ใน Model feature เยอะ ใช้เวลาเยอะ เงินเยอะ แปลผลยาก
Filter Method : เลือกที่มีความสามารถในการแยกข้อมูล ดูจาก Correlation (ข้อมูลเป็นตัวเลข) , Information Gain
Wrapper Method : เอาหลายๆ feature มารวมกันแล้วดูว่าชุดไหนดีที่สุด มี 2 วิธี
Forward Sequential Selection: กันก่อน วิธีของมันคือการที่เราเริ่มจาก Model เปล่า ๆ ก่อน แล้วค่อย ๆ ลงเติม Feature เพิ่มทีละอัน แล้ววัดผลออกมาดูว่าเป็นอย่างไร แล้วก็เลือกอันที่ดีที่สุดใส่เข้าไป ทำแบบนี้ไปเรื่อย ๆ จนครบจำนวน Feature ที่เรากำหนดไว้
Backward Elimination: ก็จะทำงานกลับกันคือ เราเลือกทั้งหมดตั้งแต่แรกเลย แล้วสิ่งที่เราต้องทำแทนที่จะเอา Feature มาใส่ เรากิโยนทิ้งแทน ก็คือ โยนอันไหนแล้วดีกว่า ก็โยนอันนั้นออกไป
Embedded Method : คือ การที่เราหยิบมันมาให้มันค้นหาค่ะ แต่เราจะให้น้ำหนักของแต่ละ Feature ไม่เท่ากัน บาง Feature คำนวณออกมา อาจจะทำให้ Weight เป็น 0 เลยก็มีเหมือนกัน (ง่าย ๆ คือ แก โดน ทิ้ง ร้ายๆ นก !) วิธีนี้เราเรียกว่า การทำ Regularisation