Chp 10 → Attribute Selection, Feature Engineering

Precision, Recall and F1 Score

$X_{n \times d}$ → n → data points
↳ attributes
↙ data set

Find relevant attributes, which are subset of $X_{n \times d}$

Why?

Ex:- Medical Dataset

Figure out Vital Signs (out of entire dataset)
→ When Smaller no of point suffice
use smaller models, faster models

Reasons
→ Interpretability
→ Faster models in Deployment
→ Better Generalization

Eg:- Stock market Data
10^6 attributes (noisy classification)
↘ 10 most important (better classification)
↙ Very Complex Loss Function
very minimal effect

How to do attribute selection?
① Exhaustive Search (all subsets)
$(10^6)^{10} \sim 10^6$

l_1-regularization

$Loss_{L2} = Loss_{original} + \lambda \Sigma w_i^2$ → regularization

Ex:-
weights [15, 16, 1, 2, 19]
↓ $l_1(-2)$
[13, 14, 0, 0, 17]

(revise)
l_1-regularization

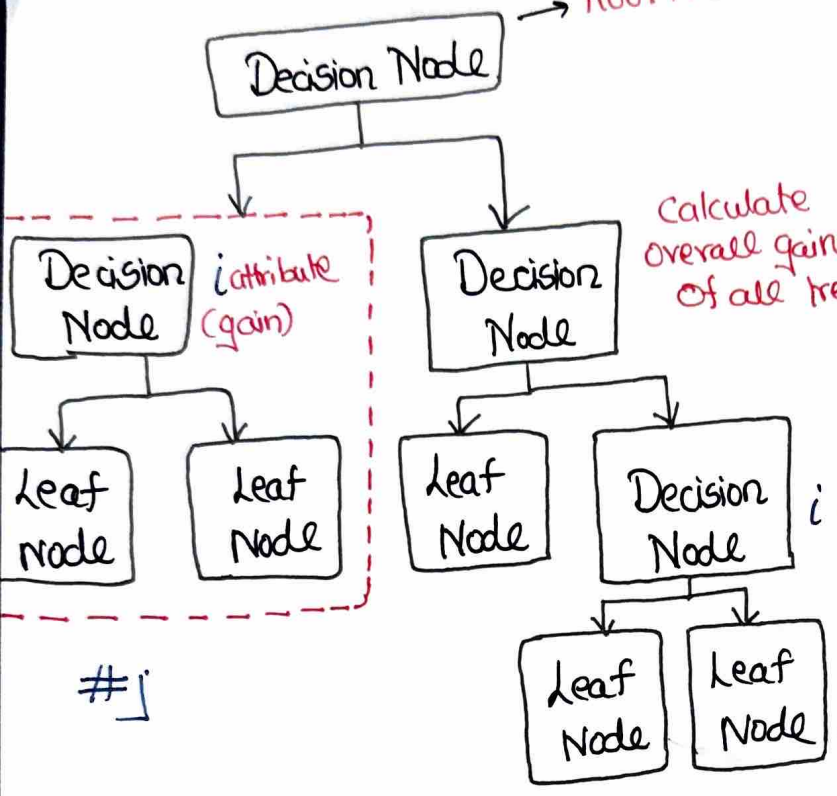
Lot of learned weights of the model close to 0
↳ those attributes are not important

(Based on Random Forest)

*Important attributes via random forests :-

(Gain) one attribute (that doesn't appear anywhere) else

*How to use random forests to do feature selection?



$g_{i,j}$ → weighted information gain for attribute 'i' when we look how its used in tree 'j'

Calculate overall gain of all trees ↗

$$G_i = \sum_{j=1}^{N} g_{i,j}$$

↳ trees (sub trees)

$$\hat{G}_i = G_i / (\sum G_i)$$ Importance scores

i → attribute (gain)

↓

normalize the scores

{ attributes which have a higher importances } considered

Random Forest → Collection of Decision Trees.

other attributes below can be repeated. There will be gain at attributes below too.

- correlated { (correlation dilutes importance) one feature can substitute another }

$$G_i = \frac{G_i}{\sum_i G_i}$$

G_i → gain of i^{th} attribute

$\sum_i G_i$ → summation of all gains across all attributes

example we have :-

attributes

Temperature (Celsius) (higher deg polynomial)

Temperature (Farenheit)

correlated

→ convey the same thing (collectively contribute high importance)

↓

each one of them would be 50% of the tree. (we may miss important attributes)

- high degree features

↳ numerical values which are bucketized may be considered important when they aren't

* Select the weakest attribute :- (Greedy Feature Elimination)

Assume for simplicity we have 4 attributes

$X_{n \times 4}$

k-nn algorithm (classification)

Classifiers	accuracy ↗
C1 : X / Attribute=1	90%
C2 : X / Attribute #2	95%
C3 : X / Attribute # 3	83%
C4 : X / Attribute # 4	96%

See their performance on train and test without the corresponding attribute

Evaluate which is the least important

Highest accuracy

C4 → without attribute #4

↓

attribute 4 → not important

(we can still reach high accuracy dropping 4)

↓

Drop 4

Ablation Testing

These classifiers are like

C1 → without attribute #1

C2 → without attribute #2

C3 → : : : #3

C4 → #4

(Greedy Feature Elimination)

↓

Requires lot of training

↓

Gives smaller dataset

$X' = X/4$

↙ Xprime

↘ X without attribute #4

If 'd' attributes

↙

we need to train 'd' times

if we want to drop 'm' attributes

↑

m times

↘ attributes

m.d → downside (computationally expensive)

Permutation Based Feature Importance :-

Solves Problems of
- Random Forest → (overestimation)
- Greedy Elimination → (computationally expensive)

- Inputs – fitted predictive model m, tabular dataset (training or validation) D
- Compute the reference scores s of the model m on data D $[n \times d]$ (for instance the accuracy for a classifier or the R^2 for a regressor)

- For each feature j (column of D):
  - For each repetition k in $1, \ldots, K$:
    - Randomly shuffle column j of dataset D to generate a corrupted version of the data named $\tilde{D}_{k,j}$
    - Compute the scores $s_{k,j}$ of model m on corrupted data $\tilde{D}_{k,j}$
  - Compute importance i_j for feature f_j defined as:

$$i_j = s - \frac{1}{K} \sum_{k=1}^{K} s_{k,j}$$

(d times / evaluation)

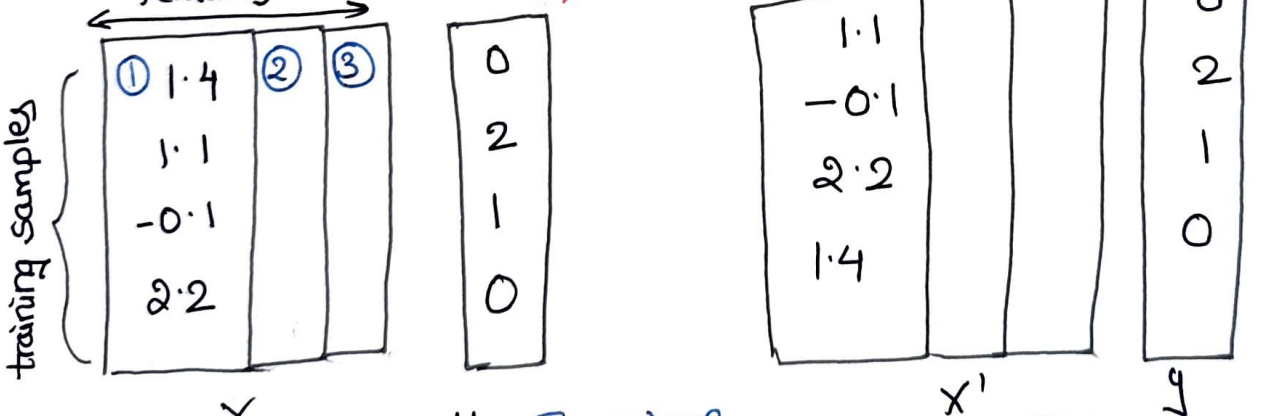
(mean of all shuffles (K))

Explanation

Accuracy = 99% (0.99)

Accuracy = 85% (0.85)

Shuffled K times



Importance (of #1) = 0.99 − 0.85 = 0.14 (higher accuracy loss) → High Importance