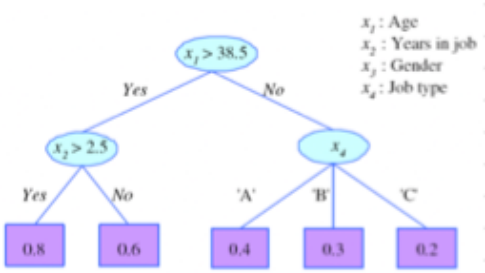
Lecture 15 - 24/10/24

Feature Importance

- Basal features are more globally important.
- Some features may not be used (feature extraction)



In this case gender is useless

Q.A.

- Are D.T. sensitive to rotation?
- How about scaling?
- How about shifting?

* Are D.T sensitive to variation in data?
  - Yes, they are, slight changes could lead to totally different trees.

Dealing with Missing Features

- We may have missing values for some features in some training sample.
  - Image each feature has 5% chance of being missing.
  - Leads to a prob of 92.3% if 50 features were sampled.

Solution → use surrogates!!!

Surrogate splits: replacement feature to be used for splitting when feature is missing.

Stop conditions:

* A node is fully pure.
* No split reduces the impurity.
* min # of data point in leaf.
* max # of nodes in the tree.
* max depth for the tree.

Variance in Decision Trees

- Decision trees can be grown to complete purity on training dataset
- It tends to memorize training set, poor generalization.
- High Variance → overfitting

* But there is a remedy → Regularization (shrinkage)
  - Early Stopping
  - Pruning

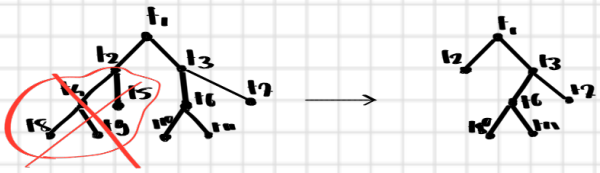
Early Stopping

- Recall the stopping conditions above!
- Stop tree from growing too large based on some stopping criteria:
  - min # of data points in a leaf
  - max # of nodes in the tree
  - max depth for the tree

Tree too restricted = High bias

Pruning → Post-pruning

- Grow a large tree, then prune it back
- ∀ subtree:
  - Collapse subtree to a leaf
  - Compare tree with subtree in terms of cost-complexity
  - If leaf (without subtree) is as good as subtree → discard subtree, keep leaf.



Pros vs Cons → of decision trees

Pros

- White boxes → very easy to explain.
- Can handle both categorical & numerical.
- Can match any distribution.
- Independent of scaling, centering.
- Inherent feature selection.
- Cheap

Cons

Not as good in prediction as other stuff

Ugly decision boundary

Sensitive to rotation!! (variation in data)

Bias/Variance problem

Ensemble Methods

The Law of Large Numbers

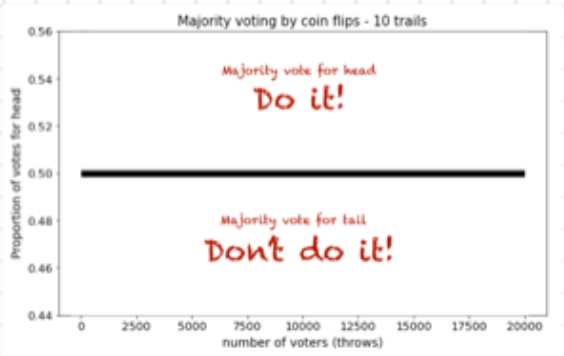
- If we sample independent & identically distributed random variables:

$$\frac{1}{m}\sum_{i=1}^{m} x_i \to \bar{x} \quad \text{as } m \to \infty$$

where $\bar{x}$ is the true expected value of the dist.

Imagine a scenario where:

Coin Toss { Head P(head) = 0.51 ; Tail P(tail) = 0.49 } ↝



We wanna decide based on majority!

Weak classifiers

- Assume not-so-good classifiers
- Individually only correct 51% of the time
- We use majority rule method.

# classifiers	# True pred.	Probability of correct prediction	Probability of correct prediction
1	1	0.51	0.51
	0	0.49	
3	3	0.51^3	
	2	$3 * 0.49 * 0.51^2$	0.515
	1	$3 * 0.51 * 0.49^2$	
	0	0.49^3	

With only 1 classifier

With 3 classifiers (at least 2 out of 3 is correct)

- With nearly 1000 classifiers, probability of correct prediction comes close to 0.75

With ~1000 classifiers

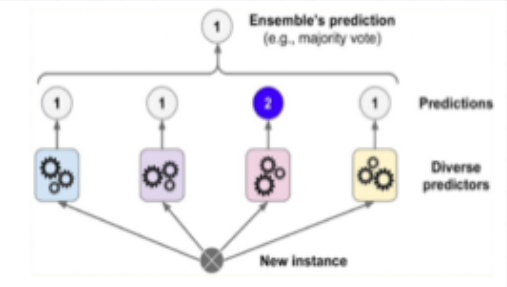
Voting Methods

There are two main voting methods:

- Hard voting (majority rule)
- Soft voting

Hard Voting

- Each classifier makes its separate prediction on the test data
- Majority rule → the final pred is the class that gets most votes!



Soft Voting

- Requires that all classifiers can estimate posterior class probs $p(k|x)$
- Final prediction is the class with highest probs, AVG over all classifiers
- Often performs better than hard voting

Example:

- We have three classifiers and two classes:

Classifier 1 estimates a posterior probability 91% for A
Classifier 2 estimates a posterior probability 49% for A
Classifier 3 estimates a posterior probability 49% for A

Hard Voting:

1 vote for A
2 votes for B
3 votes for B
→ (B)

Soft Voting:

Prob of A is:

$$\frac{(91+49+49)}{3} = 63\% \rightarrow (A)$$

Now onto ensemble methods

- A group of separate classifiers or regressors:
  - they form an ensemble (committee)
- An ensemble can perform better than any of its individual base-learners.
- Any type of ML model can be used as base-learner
- Can be used both for classification & regression.

Popular Ensembles:

Bagging Random Forest Boosting Stacking

- An ensemble mainly aims to reduce variance (law of large #)
- Another aims to reduce bias.

Reducing Variance

- As said an ensemble performs better than individuals.

  But only if base-learners are completely independent.

- Errors should be uncorrelated
- Identical learners do not increase performance.

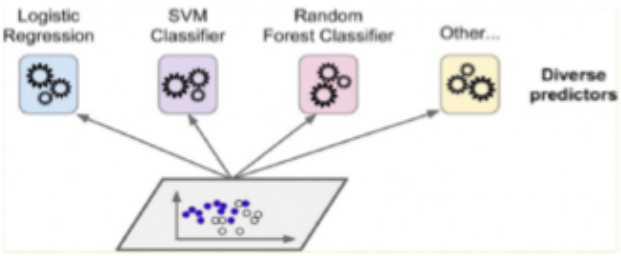
We need diversity to get closer to the independence condition!

Introducing diversity:

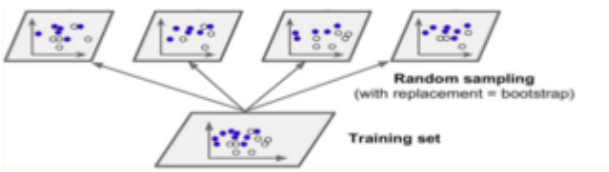
* **Diversity in predictors**
  - Heterogeneous ensemble (≠ type classifiers) trained on the same data
* **Diversity in training data**
  - Homogeneous ensemble (same type classifiers) trained on ≠ data

Diversity in the predictor

Use a heterogeneous ensemble (different types of classifiers) trained on the same dataset.



Bootstrapping



This is a way of varying (diversifying) the training data for each predictor

It is difficult to have many datasets...