FROM DECISION TREE ADABOOST

FEATURES OF THE DATASET -- Home Credit Default Risk

A Kaggle machine learning competition

Behavioral Science related (predicting whether or not a client will repay a loan or have difficulty)

Large sample size

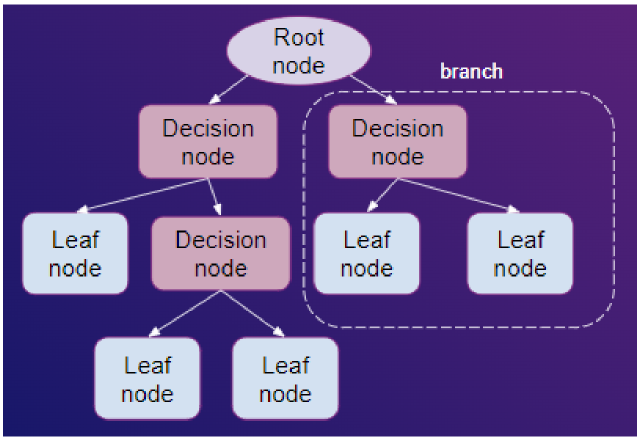
Imbalanced data

Many predictor variables

Adaboost performs better

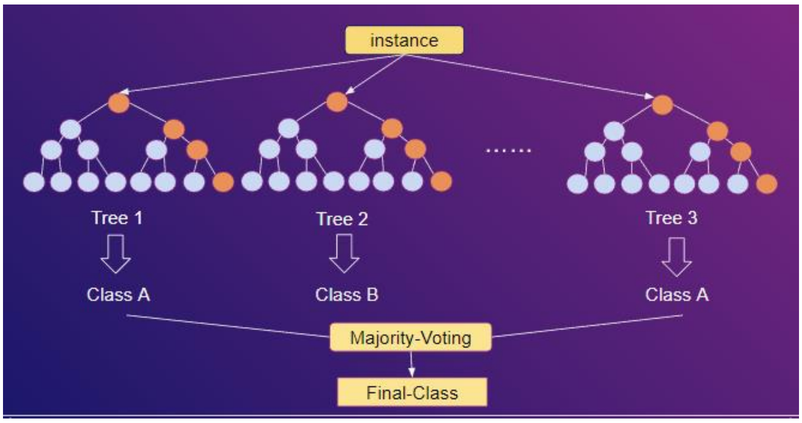
Decision Trees

In machine learning, a decision tree is a predictive model that represents a mapping relationship between attributes and values.



Tree	
Advantage	Disadvantage
Simple to understand and to interpret	Unstable
Less Data Preparation	Can cause overfitting
Able to handle both numerical and categorical data.	

Random forest



The random forest classifier combines a number of decision trees to improve the accuracy of the classification.

The final output is determined by the mode of the classes of the individual tree output.

Majority rule is a principle that means the decision-making power belongs to the group that has the most members.

Random Forest	
Advantage	Disadvantage
Relatively high accuracy	Complexity
Stable	Longer Training Period
Can process data with large number of features and samples	
Works well with both categorical and continuous variables	
Automatically handle missing values	

Application of Random Forest

Predict cardiovascular disease

Predict online buying behavior

Detect credit card fraud

Adaptive Boosting

Assumptions

1. In sample selection: Training set for adaboost is the same, only the weight of each sample is changing.
2. In the order of calculation: The classify function for adaboost must be generated sequentially.
3. In the sample weights: Adaboost adjusts the sample weights if error occurred in previous model.
4. In the prediction function: The weights for predictor function in adaboost changed based on the error rate.

The weak learners in AdaBoost are decision trees with a single split, called decision stumps.

Each stump chooses a feature, say X2, and a threshold, T, and then splits the examples into the two groups on either side of the threshold.

Sequential updating of weights on data points

Form a final model from weak learners

Random Forest	Adaboost
Chose sample for each tree	All data set were trained

Each sample has same weight	More probability to drawn misclassification sample
Majority of trees leads to the answer	Use weights when combining trees

Formula

Initialize $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$

For $t = 1$ to T:

Fit $C_t(x)$ and minimize error using weight w_i

Compute weighted error: $\epsilon_t = \boldsymbol{w}^T \, I(\boldsymbol{y} \neq C_t(x))$

Compute $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Update $w_i := w_i e^{\alpha_m I(y \neq C_t(x))}$ and normalize it

$$C(x) = sign\left(\sum_{t=1}^{T} \alpha_t C_t(x)\right)$$

$(x_1, y_1), \dots, (x_n, y_n)$, x is predictor and $y \in \{-1,1\}$ is response

t is number of iteration

$C_t(x)$ is a weak classifier trained in iteration t

w_i is the weight of observation $i \in (1, \dots, n)$

$\boldsymbol{w}$ is the column vector $[w_1 \quad w_2 \quad \cdots \quad w_{n-1} \quad w_n]^T$

α_t is the model $C_t(x)$ weighting

$I()$ is the indicator variable function (output vector for simplicity)

- Given:
  - T(X) – complexity of training for weak learner
  - t(X) – complexity of testing for weak learner
  - T – number of iteration
  - n – number of samples
  - p – number of predictors

Training Phase for Adaboost: O(TT(X)+Tn)
Testing Phase for Adaboost: O(nt(X))

Weak Learner of Decision Tree with depth = 1:
Training Phase of weak learner: T(X) = O(np)
Testing Phase of weak learner: t(X) = O(1)

Training Phase = O(Tnp)

Adaboost	
Advantages	Limitations

High precision (greatly improve the accuracy of the decision tree, comparable to SVM).	Training is time-consuming (reselect the best segmentation point for the current classifier each time).
The weight of each classifier fully considered by AdaBoost (relative to Bagging algorithm and Random Forest algorithm).	Classification accuracy drops due to data imbalance.
Various methods to build sub-classifiers (AdaBoost provides a framework).	The number of AdaBoost iterations (i.e.he number of weak classifiers) is not easy to set. Cross-validation can be used to make the determination.
Good use of weak classifiers for cascading.	Sensitive to noisy data and anomalous data.
Simple, efficient, easy to write and almost no overfitting.	
No parameters to adjust during the training process.	

Application of Adaboost

For binary or multi-category scenarios

Baseline for classification tasks (simple, no overfitting, no need to adjust the classifier)

For feature selection (feature selection)

Correction the bad case (only need to add a new classifier, no need to change the original classifier)