Binary Classification Methods

05 October 2022 1:25 AM

Methods of modelling binary classification

Non-applicable points are crossed out

Method	Advantages	Limitations + Considerations
Logistic Regression (LR) OPTION 1	• Simple • Weights are Interpretable • ~~Less prone to over-fitting for low dimensional~~ • Efficient in handling linear features • Low computational time	• Prone to over-fitting on high dimensional data • Cannot handle non-linear problems; most IRL data are • Cannot capture complex • Susceptible to multicollinearity • Requires independent variables that are linearly • Need to remove unimportant • Sensitive to outliers
Lasso: Will set some coefficients to zero.	• Reduces overfitting by eliminating features,	• Does not work well with multicollinearity - will randomly select one of them • Decreased flexibility - May cause small bias when eliminate too much variables
Ridge: Will set coefficients close to zero	• Reduces impact of not important features in predicting target, compared to LR • Decreases variance (at the cost of bias)	• Does not eliminate coefficient
Elastic Net: Combination of ridge and lasso OPTION 2	• Has both feature elimination (lasso) and feature reduction (ridge), compared to LR - Ridge prevents lasso from eliminating too many • Does not easily eliminate multicollinear variables	• More computationally costly - need to cross-validate the weight of L1 & L2 penalty • Greater flexibility than lasso - increased probability of
Decision Trees (DT)	• Can capture non-linear relationship • Does not require feature transformation for non- • Does not require normalization • Does not require feature scaling (= scale- • Interpretable + can be visualized • Can handle numerical, categorical, and Boolean • Not sensitive missing values • Not sensitive to outsider data (= stable) • Non-parametric - does not require assumption	• Usually for multi-class • Requires more training time with many features • Requires pruning to mitigate overfitting • Tends to overfit • Bad for big data - overfitting and high complexity • Does not guarantee 100% efficient decision tree
Random forest Not good	• Handles non-linear parameters efficiently • Works well with categorical and continuous • Can automatically handle missing values • Scale-invariant • Robust to outliers • Stable • Less impacted by noise compared to DT • Higher accuracy than DT • Less prone to overfitting than DT • Can product feature importance - can be used	• Requires a lot of • Longer training period than DT • Altho provide feature importance, it is not easy to • Susceptible to multicollinearity

	for feature selection • Can perform feature selection • Good for imbalanced dataset	
KNN Not good	• Does not require training • No training period - time efficient • Can add new data • Easy to implement • Only has one hyperparameter • Choice of distance metric • Can learn non-linear decision boundaries • No assumptions	• Does not work well with large • Does not work well with high dimensions • Needs feature scaling • Sensitive to noisy data, missing values, and outliers • Requires hyperparameter • Does not perform well on imbalanced data • Assigns equal weight to every
Neural Networks Not good	• Can capture complex relationships • Good for non-linear data • Good for large number of inputs • Good for large number of features • Once trained, predicts fasts • No assumptions	• Black boxes - cannot interpret features • Computationally expensive • Time consuming • ~~Depends on a lot of training~~
SVM	• Good for non-noisy target variable i.e. binary • Good for high dimensional data • Less sensitive to outliers	• Not suitable for large dataset due to time • Difficult to select the right kernel • Hyperparameter optimization is important for generalization • Susceptible to multicollinearity
Gradient Boosting (https://neptune.ai/blog/gradient-boosted-decision-trees-guide)	• More accurate • Faster on large dataset • Supports categorical features • Can handle missing values natively	• Prone to overfitting - can be solved using L1 & L2 penalties or low learning rate • Computationally and time • Hard to interpret final models
e.g. XGBoost OPTION 3	• Less feature engineering - No need scaling or • Can obtain feature importance - can be used for feature selection • Not sensitive to outliers or missing values, • Fast to interpret • Good execution speed • Good model performance • Less prone to overfitting Note: Good for classification problems with many features and a large dataset	• Difficult to interpret; tough to visualize • Parameters must be tuned properly to prevent overfitting • Harder to tune with more hyperparameters
Naive Bayes Classifier (?)	• Very fast • Insensitive to irrelevant features • Scalable to large dataset • Good with high dimensions	• Requires each features to be independent - difficult with irl • Bad estimator - (not good for interpretability?) • Training data should represent population
Perceptron (?)		

Bayesian networks (?)		

Note:
- Lasso, ridge, and elastic net can be used for classification by using deviance instead of RSS

Sources
- https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/
- https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04
- https://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-knn.html
- https://www.universelnews.com/2022/05/06/advantages-disadvantages-of-knn-in-machine-learning/
- https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6

Methods for building decision trees

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/

3. Available algorithms and software packages for building decision tree models Go to: ▸

Several statistical algorithms for building decision trees are available, including CART (Classification and Regression Trees), [7] C4. 5, [8] CHAID (Chi-Squared Automatic Interaction Detection), [9] and QUEST (Quick, Unbiased, Efficient, Statistical Tree). [10] Table 1 provides a brief comparison of the four most widely used decision tree methods. [11,12]

Table 1.

Comparison of different decision tree algorithms

Methods	CART	C4. 5	CHAID	QUEST
Measure used to select input variable	Gini index; Twoing criteria	Entropy info-gain	Chi-square	Chi-square for categorical variables; J-way ANOVA for continuous/ordinal variables
Pruning	Pre-pruning using a single-pass algorithm	Pre-pruning using a single-pass algorithm	Pre-pruning using Chi-square test for independence	Post-pruning
Dependent variable	Categorical/ Continuous	Categorical/ Continuous	Categorica	Categorical
Input variables	Categorical/ Continuous	Categorical/ Continuous	Categorical/ Continuous	Categorical/ Continuous
Split at each node	Binary; Split on linear combinations	Multiple	Multiple	Binary; Split on linear combinations

Open in a separate window

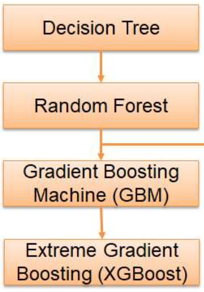
Other
- https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

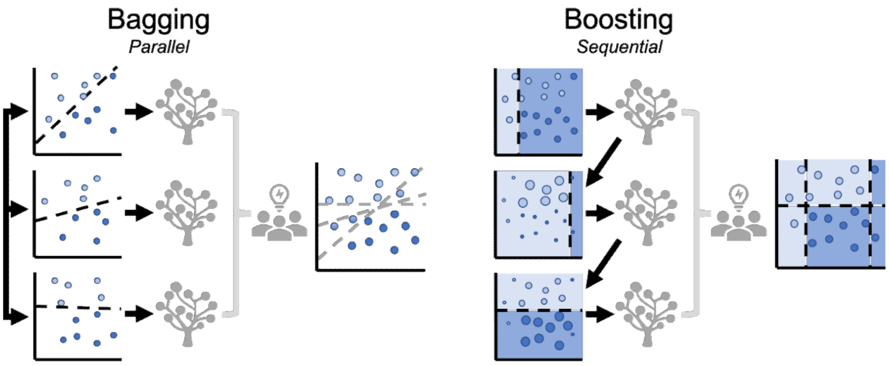
https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

Algorithms used in Decision Trees
- ID3 → (extension of D3)
    - ID3 algorithm builds decision trees using a top-down greedy search approach through

the space of possible branches with no backtracking
    - A greedy algorithm, as the name suggests, always makes the choice that seems to be the best at that moment.
- C4.5 → (successor of ID3)
- CART → (Classification And Regression Tree)
- CHAID → (Chi-square automatic interaction detection Performs multi-level splits when computing classification trees)
- MARS → (multivariate adaptive regression splines)

https://www.analyticsvidhya.com/blog/2021/04/distinguish-between-tree-based-machine-learning-algorithms/
- Ensemble methods: Bagging (Random Forest), Boosting (Adaptive Boosting, GBM & XGBoost) and Stacking





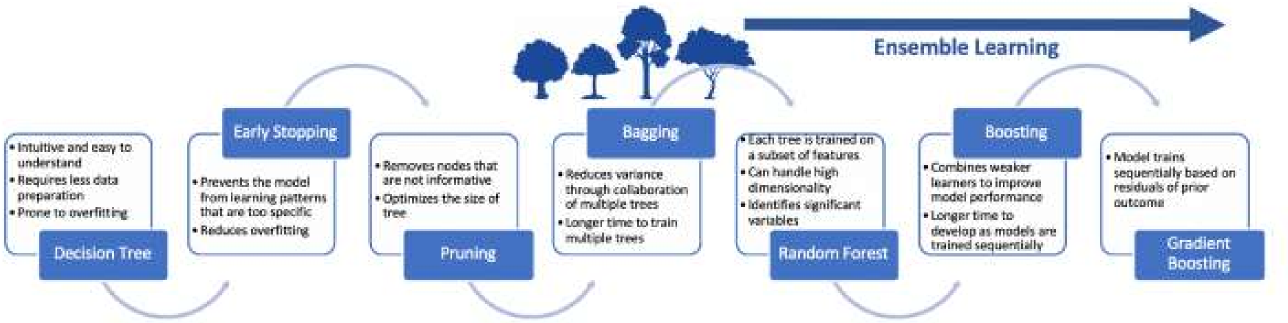
Note:
- Gradient Boosting Machine (GBM) creates trees from residual datapoints
- Extreme Gradient Boosting (XGBoost) is a regularized form of GMB to control overfitting
- There is also LightGBM and CatBoost
    - https://neptune.ai/blog/xgboost-vs-lightgbm
    - https://neptune.ai/blog/gradient-boosted-decision-trees-guide
    - https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm

https://neptune.ai/blog/gradient-boosted-decision-trees-guide
https://hackernoon.com/boosting-algorithms-adaboost-gradient-boosting-and-xgboost-f74991cad38c
Boosting algorithms in ML
- Gradient boosting - an ensemble of weak learners is used to improve the performance of a machine learning model.
- Adaptive Boosting (AdaBoost) - AdaBoost fits a sequence of weak learners to the data. It then assigns more weight to incorrect predictions, and less weight to correct ones.
- XGBoost - uses regularization on gradietn boosting
- LightGBM - uses a leaf-wise tree growth algorithm
    - Converge faster, but more prone to overfitting
- CatBoost - grows a balanced tree using oblivious decision trees (uses the same features to make the right and left split at each level of the tree).

https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm
CatBoost

1. Symmetric trees --> Time and computationally efficient, and regularizes
2. Ordered boosting - a permutation-driven approach to train model on a subset of data while calculating residuals on another subset
3. Native feature support - Can handle all data type --> saves time pre-processing

https://towardsdatascience.com/the-evolution-of-trees-based-classification-models-cb40912c8b35#:~:text=Tree%2Dbased%20classification%20models%20are,be%20used%20to%20make%20predictions.
Evolution of tree-based models:



Comparing ML algorithms: https://neptune.ai/blog/how-to-compare-machine-learning-models-and-algorithms - This is pretty good