2025

4Geeks Academy: data science cohort 12

DAY 22: GRADIENT BOOSTING

TODO

GRADIENT BOOSTING

Model details, applications and hyperparameters

RANDOM FOREST PROJECT

Submit Random Forest Project Tutorial (Random Forest Algo. module), if you haven't done so already

GRADIENT BOOSTING PROJECT

Work on Boosting Algorithms Project Tutorial (Gradient Boosting Algo. module), plan to finish before class Friday

TOPICS

01 GRADIENT BOOSTING MODELS

02 APPLICATIONS

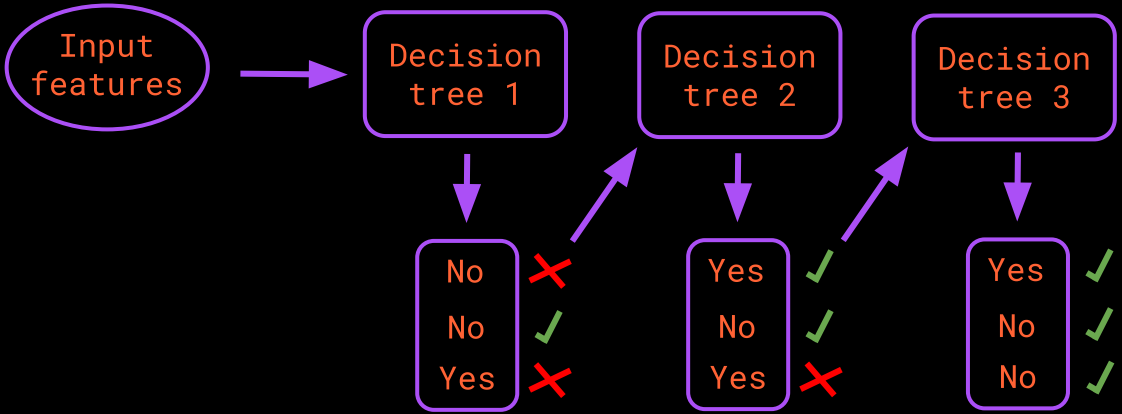
03 HYPERPARAMETERS

RANDOM FOREST MODEL

WHAT Ensemble of decision trees, where each tree learns the prior tree’s mistakes

WHY More powerful than decision trees or random forests

HOW



APPLICATIONS

IMPLEMENTATIONS

- **Scikit-learn**
  - Has classification and regression variants
  - Has ‘normal’ and histogram implementation
  - Uses familiar API
- **XGBoost (DMCL)**
  - GPU support
  - Distributed training support (Spark)
  - Generally more options/features than sklearn
- **LightGBM (Microsoft)**
  - GPU support
  - Distributed training
  - Fast histogram based implementation

PROS

- More powerful than decision trees or random forests

CONS

- More computationally expensive
- More prone to overfitting
- Less interpretable

HYPERPARAMETERS (sklearn)

ENSEMBLE

- `n_estimators`: number of individual trees to build
- `learning_rate`: shrinkage factor for contributions of each additional tree
- `n_iter_no_change`: early stopping - off by default

TREE

- `max_depth`: how many splits deep will the tree go?
- `min_samples_split`: minimum sample remaining in a leaf to keep splitting
- `max_features`: maximum features to consider for splitting at each node
- `max_leaf_nodes`: maximum number of leaf nodes to create
- `min_impurity_decrease`: minimum gain in score to split

Optimizing these parameters can still help with overfitting!