# Machine-Learning-with-Tree-Based-Models-in-Python

PART 1 - CLASSIFICATION AND REGRESSION TREES (CART)

- Set of supervised learning models used for problems involving classification and regression

Classification Tree

- Sequence of if-else questions about individual features

- Objective: Infer class labels

- Able to capture non-linear relationships between features and labels

- Don't require feature scaling. For example, it do not need standardization etc.

Decision regions

- Decision region is the region in the feature space where all the instances are assigned to one class label.

- For example, if result is of two class Pass or Fail. Then there will be 2 decision region. One is Pass region other is Fail region.

Decision boundary

- It is the seperating boundary between two region.

- In above example, decision boundary will be 33% (which is the passing marks)

Logistic regression vs classification tree

- A classification tree divides the feature space into rectangular regions.

- In contrast, a linear model such as logistic regression produces only a single linear decision boundary dividing the feature space into two decision regions.

- In other word, decision boundary produced by logistic regression is linear (line) while the boundaries produced by the classification tree divide the feature space into rectangular regions (Not a line but boxes/region it divides two class).

Building block of Decision Tree

- Root: No parent node, question giving rise to two children nodes.

- Internal node: One parent node, question giving rise to two children nodes.

- Lead: One parent node, no children node -> Prediction.

Classication-Tree Learning (Working) -

- Nodes are grown recursively (based on last node).

- At each node, split the data based on:

1. feature f and split-point(sp) to maximize IG(Information gain from each node).

2. If IG(node)= 0, declare the node a leaf.

Information Gain-

- IG is a synonym for Kullback–Leibler divergence.

- It is the amount of information gained about a random variable or signal from observing another random variable.

- The term is sometimes used synonymously with mutual information, which is the conditional expected value of the Kullback–Leibler divergence.

- KL divergance is the univariate probability distribution of one variable from the conditional distribution of this variable given the other one.

Criteria to measure the impurity of a node I(node):

1. Variance (Regression) [Variance reduction of a node N is defined as the total reduction of the variance of the target variable x due to the split at this node]

2. Gini impurity (Classification) [Measure of impurity. Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset]

3. Entropy (Classification) [Measure of purity. Information entropy is the average rate at which information is produced by a stochastic source of data]

Note

- Most of the time, the gini index and entropy lead to the same results.

- The gini index is slightly faster to compute and is the default criterion used in the DecisionTreeClassifier model of scikit-learn

Regression Tree Classification

- Measurement are done through MSE (Mean Square error)

- Information Gain is the MSE. So the target variable will have the Mean Square Error.

- Regression trees tries to find the split that produce the leaf where in each leaf, the target value are an average of closest possible to the mean value of labels in that leaf.

PART 2 - BIAS VARIANCE TRADEOFF

Supervised Learning

- y = f(x), f is the function which is unknown

- Our model output will be that function

- But that function may contains various type of error like noise

Goals of Supervised Learning

- Find a model f1 that best approximates f: f1 ≈ f ()

- f1 can be LogisticRegression, Decision Tree, Neural Network ...

- Discard noise as much as possible.

- End goal:f1 should acheive a low predictive error on unseen datasets.

Difculties in Approximating f

- Overtting: f1(x) fits the training set noise.

- Undertting: f1 is not flexible enough to approximate f

Generalization error

- Generalization Error of f1 : Does f1 generalize well on unseen data?

- It can be decomposed as follows: Generalization Error of

- f1 = bias + variance + irreducible error

Bias

- Bias: error term that tells you, on average, how much f1 ≠ f.

- High Bias lead to underfitting

Variance

- Variance: tells you how much f is inconsistent over different training sets.

- High Variance lead to overfitting

- If we decrease Bias then Variance increase. Or Vice versa.

Model Complexity

- Model Complexity: sets the flexibility of f1.

- Example: Maximum tree depth, Minimum samples per leaf etc etc.

Bias Variance Tradeoff

- It is the problem is in trying to simultaneously minimize these two sources of error that prevent supervised learning algorithms from generalizing beyond their training set.

Estimating the Generalization Error, Solution:

- Split the data to training and test sets

- Fit t1 to the training set

- Evaluate the error of f1 on the unseen test set

- Generalization error of f1 ≈ test set error of f1.

Better Model Evaluation with Cross-Validation

- Test set should not be touched until we are confident about f1's performance.

- Evaluating f1 on training set: biased estimate,f1 has already seen all training points.

- Solution → K Cross-Validation (CV)

Diagnose Variance Problems

- If f1 suffers from high variance: CV error of f1 > training set error of f1.

- f1 is said to overfit the training set. To remedy overtting:

- decrease model complexity

- for ex: decrease max depth, increase min samples per leaf

- gather more data

Diagnose Bias Problems

- If f1 suffers from high bias: CV error of f1 ≈ training set error of f1 >> desired error.

- f1 is said to underfit the training set. To remedy underfitting:

- increase model complexity

- for ex: increase max depth, decrease min samples per leaf

- gather more relevant features

Limitations of CARTs

- Classication: can only produce orthogonal decision boundaries.

- Sensitive to small variations in the training set.

- High variance: unconstrained CARTs may overt the training set.

- Solution: ensemble learning.

Ensemble Learning

- Train different models on the same dataset.

- Let each model make its predictions.

- Meta-model: aggregates predictions of individual models.

- Final prediction: more robust and less prone to errors.

- Best results: models are skillful in different ways.

Steps in Ensemble learning

1. Training set is fed to different classifier like Decision tree, Logistic regression, KNN etc.

2. Each classifier learn its parameter and make prediction

3. Each prediction are fed into another model and that model make final prediction.

4. That final model is known as ensemble model.

PART 3 - BAGGING AND RANDOM FOREST

Bagging

- Bagging is an ensemble method involving training the same algorithm many times using different subsets sampled from the training data

- In bagging, it uses same algorithim (only one algo is used)

- However the model is not training on entire training set

- Instead each model is trained on different subset of data

- Bagging: Bootstrap Aggregation.

- Uses a technique known as the bootstrap.

- Reduces variance of individual models in the ensemble.

- For example, suppose a training dataset contains 3 parts - a,b,c.

- It create subset by method sample by replacement. For example aaa,aab,aba,acc,aca etc.

- On this subset, the models are trained.

Bagging Classication

- Aggregates predictions by majority voting (Final model is selected by voting).

- BaggingClassifier in scikit-learn.

Bagging Regression

- Aggregates predictions through averaging (Final model is selected by avergaing).

- BaggingRegressor in scikit-learn.

Bagging limitations

- Some instances may be sampled severaltimes for one model,

- Other instances may not be sampled at all.

Out Of Bag (OOB) instances

- On average,for each model, 63% ofthe training instances are sampled.

- The remaining 37% constitute the OOB instances.

- Since OOB instances are not seen by the model during training.

- This can be used to estimate the performance of the model without the need of cross validation.

- This technique is known as OOB evaluation

Random Forest

- Another ensemble model

- Base estimator: Decision Tree

- Each estimator is trained on a different bootstrap sample having the same size as the training set

- RF introduces further randomization in the training of individual trees

- d features are sampled at each node without replacement ( d < total number of features)

Random Forests Classication:

- Aggregates predictions by majority voting

- RandomForestClassifier in scikit-learn

Random Forests Regression:

- Aggregates predictions through averaging

- RandomForestRegressor in scikit-learn

Feature Importance

- Tree-based methods: enable measuring the importance of each feature in prediction.

- In sklearn :

- how much the tree nodes use a particular feature (weighted average) to reduce impurity

- accessed using the attribute feature\_importance\_

Part 4 BOOSTING

- Boosting refers to an ensemble method in which several models are trained sequentially with each model learning from the errors of its predecessors.

- Boosting: Ensemble method combining several weak learners to form a strong learner.

- Weak learner: Model doing slightly better than random guessing.

- Example of weak learner: Decision stump (CART whose maximum depth is 1).

- Train an ensemble of predictors sequentially.

- Each predictor tries to correct its predecessor.

- Most popular boosting methods: AdaBoost, Gradient Boosting.

Adaboost

- Stands for Adaptive Boosting.

- Each predictor pays more attention to the instances wrongly predicted by its predecessor.

- Achieved by changing the weights of training instances.

- Each predictor is assigned a coefficient α.

- α depends on the predictor's training error

- Learning rate: 0 < η ≤ 1. It help to shrink coeeficient α. It is the tradeoff between η and number of estimator.

- Smaller number of η should be compensiated by high number of estimator.

AdaBoost Classication

- Weighted majority voting.

- In sklearn: AdaBoostClassifier.

AdaBoost Regression

- Weighted average.

- In sklearn: AdaBoostRegressor.

Gradient Boosted Trees

- Sequential correction of predecessor's errors.

- Does not tweak the weights of training instances.

- Fit each predictor is trained using its predecessor's residual errors as labels.

- Gradient Boosted Trees: a CART is used as a base learner.

Gradient Boosted Regression:

- y = y + ηr + ... + ηr

- In sklearn: GradientBoostingRegressor .

Gradient Boosted Classication:

- In sklearn: GradientBoostingClassifier .

Gradient Boosting: Cons

- GB involves an exhaustive search procedure.

- Each CART is trained to find the best split points and features.

- May lead to CARTs using the same split points and maybe the same features.

Stochastic Gradient Boosting

- Each tree is trained on a random subset of rows of the training data.

- The sampled instances (40%-80% ofthe training set) are sampled without replacement.

- Features are sampled (without replacement) when choosing split points.

- Result: further ensemble diversity.

- Effect: adding further variance to the ensemble oftrees.

PART 5 - MODEL TUNING

- The hyperparameters of a machine learning model are parameters that are not learned from data.

- They should be set prior to fitting the model to the training set.

Parameters

- learned from data

- CART example: split-point of a node, split-feature of a node, ...

Hyperparameters

- not learned from data, set prior to training

- CART example: max\_depth , min\_samples\_leaf , splitting criterion ...

What is hyperparameter tuning?

- Problem: search for a set of optimal hyperparameters for a learning algorithm.

- Solution: find a set of optimal hyperparameters that results in an optimal model.

- Optimal model: yields an optimal score.

- Score: in sklearn defaults to accuracy (classication) and R-squared (regression).

- Cross validation is used to estimate the generalization performance.

Why tune hyperparameters?

- In sklearn, a model's default hyperparameters are not optimal for all problems.

- Hyperparameters should be tuned to obtain the best model performance.

Approaches to hyperparameter tuning

- Grid Search

- Random Search

- Bayesian Optimization

- GeneticAlgorithms etc.

Grid search cross validation

- Manually set a grid of discrete hyperparameter values.

- Set a metric for scoring model performance.

- Search exhaustively through the grid.

- For each set of hyperparameters, evaluate each model's CV score.

- The optimal hyperparameters are those ofthe model achieving the best CV score.

Grid search cross validation: example

- Hyperparameters grids:

- max\_depth = {2,3,4},

- min\_samples\_leaf = {0.05, 0.1}

- hyperparameter space = { (2,0.05) , (2,0.1) , (3,0.05), ... }

- CV scores = { score , ... }

- optimal hyperparameters = set of hyperparameters corresponding to the best CV score.

Tuning a RF's Hyperparameters

- Tuning is expensive

Hyperparameter tuning:

- computationally expensive,

- sometimes leads to very slight improvement,

- Weight the impact oftuning on the whole project.