SNS COLLEGE OF ENGINEERING
(Autonomous)
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Artificial Intelligence & Machine Learning

Decision tree algorithm in machine learning – Ensemble methods

Prepared by,
P.Ramya
Assistant Professor/ECE
SNS College of Engineering

Decision Tree

- Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter.

Pros of Decision Trees

- Decision Trees can be used for regression or classification, though they are more popular for classification problems. Generally, if you want to use a decision tree for a regression model, you should use an ensemble method.
- Decision Trees are non-parametric, which is just a fancy way to say that we aren't making any assumptions about how our data is distributed and our model's structure (parameters) will be determined from user input and the observations in our sample, rather than being fixed from the data. Non-parametric models are great when we have a lot of data, but not too much knowledge surrounding the data.

Contd…

- Decision Trees are interpretable, which means that after we build the model, we can also make inferences about our data, not just predictions. In sklearn, we can do this using the feature_importances_ attribute.
- Decision Trees are fast because they're simple and "greedy". While we might not prioritize speed when building a final, deployable model, this can be a huge bonus if we're just trying to build a model to get an initial understanding of our data

Contd...

- Finally, data preprocessing is easier with Decision Trees because we don’t have to scale our data — the splits that occur at each node are worried about one feature at a time! Depending on the algorithm used for determining split, categorical features may not have to be encoded numerically either.

Cons of Decision Trees

- The most significant disadvantage of Decision Trees is that they are prone to overfitting. Decision Trees overfit because you can end up with a leaf node for every single target value in your training data.
- Decision Trees are also locally optimized, or greedy, which just means that they don't think ahead when deciding how to split at any given node.
- Because of the greedy nature of splitting, imbalanced classes also pose a major issue for Decision Trees when dealing with classification. At each split, the tree is deciding how to best split up classes into the next two nodes. So when one class has very low representation (the minority class), many of those observations can get lost in the majority class nodes, and then prediction of the minority class will be even less likely than it should, if any nodes predict it at all.

Ensemble Methods

Ensemble methods are a fantastic way to capitalize on the benefits of decision trees, while reducing their tendency to overfit. However, they can get pretty complex and can turn into black box models. One of the best things we can do as data scientists and machine learning engineers is make sure we know what's really going on under the hood when we call that sklearn .fit() method.

Contd...

Some of the most popular ensemble methods based on Decision Trees are:

- Random Forest (Regressor / Classifier)
- Extremely Randomized Trees (Regressor / Classifier)
- Bagging (Regressor / Classifier)
- Adaptive Booster (Regressor / Classifier)
- Gradient Boost (Regressor / Classifier)
- XGBoost (Regressor / Classifier)


thank
you