Essential Notes for Decision Tree Modeling

Utsab

August 12, 2025

1. Data Preparation

- **Handle Categorical Variables:** Use one-hot encoding (e.g., `pd.get_dummies`) to avoid implying any order where none exists.
- **No Scaling/Normalization Needed:** Decision Trees split based on thresholds, so feature scaling is irrelevant.
- **Missing Values:** Scikit-learn's DecisionTree does not handle missing values directly. Impute using `SimpleImputer`.
- **Feature Types:** Trees can handle both numerical and categorical (encoded) features.

2. Key Hyperparameters

- `max_depth`: Maximum depth of the tree (controls complexity).
- `min_samples_split`: Minimum samples required to split a node.
- `min_samples_leaf`: Minimum samples at a leaf node.
- `criterion`: `"gini"` (default, faster) or `"entropy"` (information gain-based).
- `max_features`: Number of features to consider when splitting.

Rule: If we do not limit depth or splits, the tree will overfit.

3. Avoiding Overfitting

- Trees tend to memorize training data.
- Prune the tree by limiting depth or minimum samples.
- Check train vs test accuracy gap: a large gap indicates overfitting.
- Use cross-validation (`cross_val_score`) to validate performance.

4. Model Evaluation

- **Classification:** Accuracy, Precision, Recall, F1-score, Confusion matrix.
- **Regression:** RMSE, MAE, R^2 score.
- Run with different `random_state` values to check stability.

5. Interpretability

- Feature Importance: `model.feature_importances_` shows key drivers.
- Visualization: Use `plot_tree` or Graphviz for debugging and explaining results.
- Keep trees shallow for human interpretability.

6. Computational Considerations

- Decision Trees are fast to train, but large datasets with many features can lead to deep trees and high memory use.
- For high-dimensional data, consider Random Forest or Gradient Boosting for better generalization.

7. When to Use Decision Trees

Good for:

- Interpretable models.
- Mixed categorical and numerical features.
- Non-linear decision boundaries.

Avoid when:

- Dataset is extremely small (prone to overfitting).
- We need smooth, continuous decision boundaries (trees are axis-aligned).