Machine Learning (ML)

Chapter 11:

Decision Trees and Random Forest algorithm

Saeed Saeedvand, Ph.D.

Outline

In this Chapter:

- ✓ What is the Decision Tree algorithm
- ✓ Different Splitting Measures for Decision Tree algorithm
- ✓ How to find thresholds and best feature
- ✓ Advantages and disadvantages of Decision Tree algorithm
- ✓ Example of Decision Tree algorithm for both classification and regression task
- ✓ Understand Random Forest algorithm

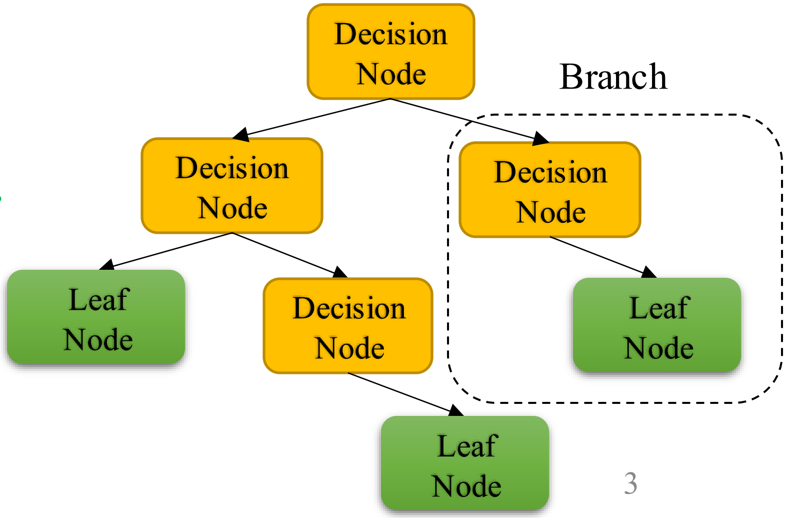
Aim of this chapter:

- ✓ Understanding the Decision Tree algorithm and Random Forest algorithm with seeing implementation examples for both.

Decision Trees

What is the Decision Tree?

- ✓ Decision tree algorithms are popular supervised ML algorithm for both classification and regression tasks.
- ✓ They widely used because of simplicity, interpretability, and since they work on non-linear data.
- ✓ Decision tree algorithms are tree-like models:
  - Each branch represents a decision rule.
  - Each decision internal node represents a feature.
  - Each leaf node represents an outcome.



Decision Trees

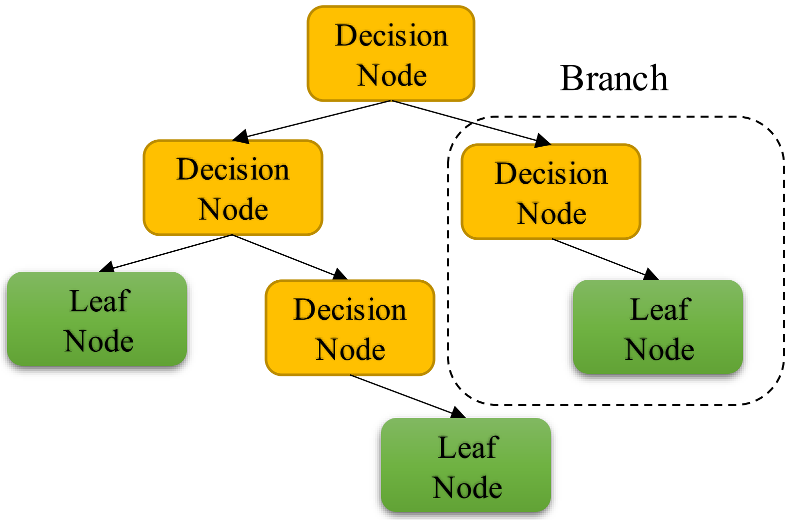
What is the Decision Tree?

Therefore, another definition is a flowchart-like structure:

✓ Branch represents the outcome of the test

✓ Decision node tests a feature attribute

✓ Leaf node represents a class label (classification) or a numeric value (regression).



Decision Tree

Decision Tree Learning

- ✓ We construct Decision Trees using a greedy top-down approach called recursive partitioning.

- ✓ Single tree can include different features at different internal nodes, but usually we run feature selection first.

- ✓ We perform recursive partitioning until a stopping criterion is met:
  - Reaching a maximum depth.
  - Reach a minimum number of samples at each leaf.

Decision Tree

Different Splitting Measures

Regression tasks

- ✓ Mean Squared Error (MSE)
- ✓ Mean Absolute Error (MAE)
- ✓ …

Classification tasks

- ✓ **Gini impurity:** Measures the probability of misclassifying a randomly chosen data from the dataset.

$$\text{Gini impurity} = 1 - \underbrace{\sum}_{\text{summation over all classes}} \underbrace{{p_i}^2}_{\text{probability of belonging to the class i}}$$

- ✓ **Entropy:** Measures the level of uncertainty in a set of examples based on class label distributions.

$$\text{Entropy} = -\sum (p_i * \log_2 p_i)$$

Decision Tree

Decision Tree Algorithm

1. **Splitting Measure**:
   - ✓ We select a splitting measure, such as MSE, Gini impurity or entropy, that measures the information content of the data at a specific node.

2. **Determine the Best Split for <u>Each Feature</u> (independent variable):**

   A) Evaluate Possible Thresholds:
   - For each feature, consider a range of potential thresholds that can be used to split the data.

   B) Calculate Impurity or Information Gain by selected splitting measure:
   - After splitting the data based on a threshold, calculate the impurity gain each part.

   C) Find the Best Threshold:
   - Chose the threshold and feature that reduces the Impurity gain as (minimizes).

Decision Tree

Decision Tree Algorithm

3. **Split the Data**:
   - ✓ Use the determined threshold to split the data into two groups.

4. **Recursive Partitioning**:
   - ✓ Repeat the above steps recursively for each child node, also considering the remaining features, until a stopping criterion is met, e.g.:
     - Reaching a maximum depth
     - A minimum number of samples per leaf

5. **Assigning Class Labels:**
   - ✓ Once the tree is constructed, assign class labels to the leaf nodes based on majority voting or other techniques.

Decision Tree

How to find thresholds and best feature?

- ✓ The decision tree algorithm evaluates all features and their potential thresholds to find the best split based on the chosen metric (criterion).
- ✓ This can be minimizing impurity or maximizing of the information gain.

This is NP-complete problem!

- ✓ Instead we use various strategies to find good splits efficiently such as:
  - **Greedy Approach**: locally optimal decisions at each node without considering the global optimum (usually this is the case).
  - **Heuristics**: consider a subset of values, …
  - **Optimizations**: Grid Search, …

Decision Tree

Handling Continuous Features:

Strategies for handling continuous features in Decision Trees:

Binary splits:

- ✓ Create binary decision rules by comparing feature values to a threshold, (usually manually defined in advance).

Threshold-based splits:

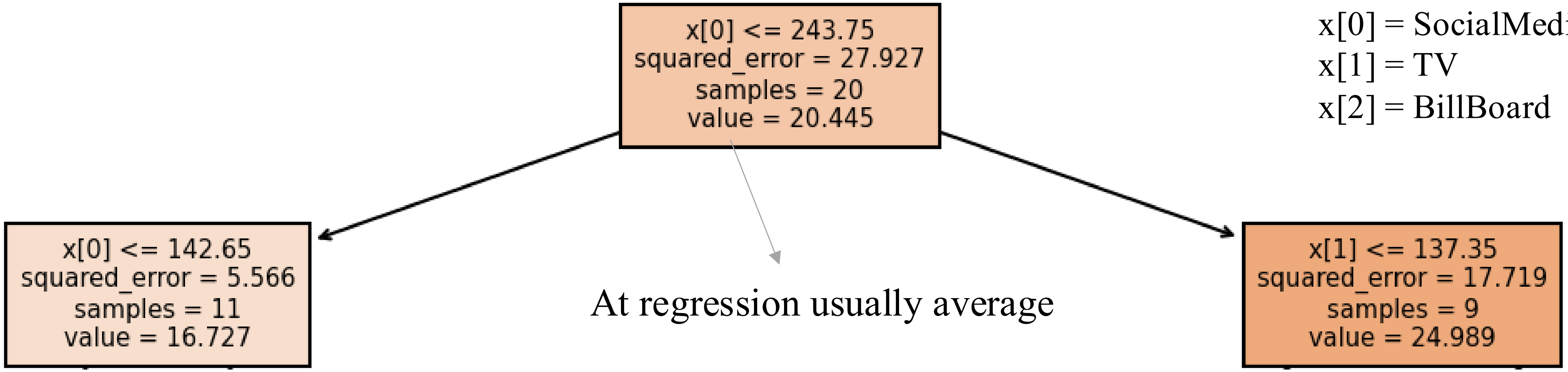
- ✓ Determine the best threshold to split the continuous feature by evaluating different thresholds, (determine the optimal threshold by evaluating different threshold values)

Note: both finally doing same thing by comparing feature with a threshold.

Decision Tree - Example

Regression

✓ Based on the advertisement example if we run decision tree regression:



x[0] = SocialMedia
x[1] = TV
x[2] = BillBoard

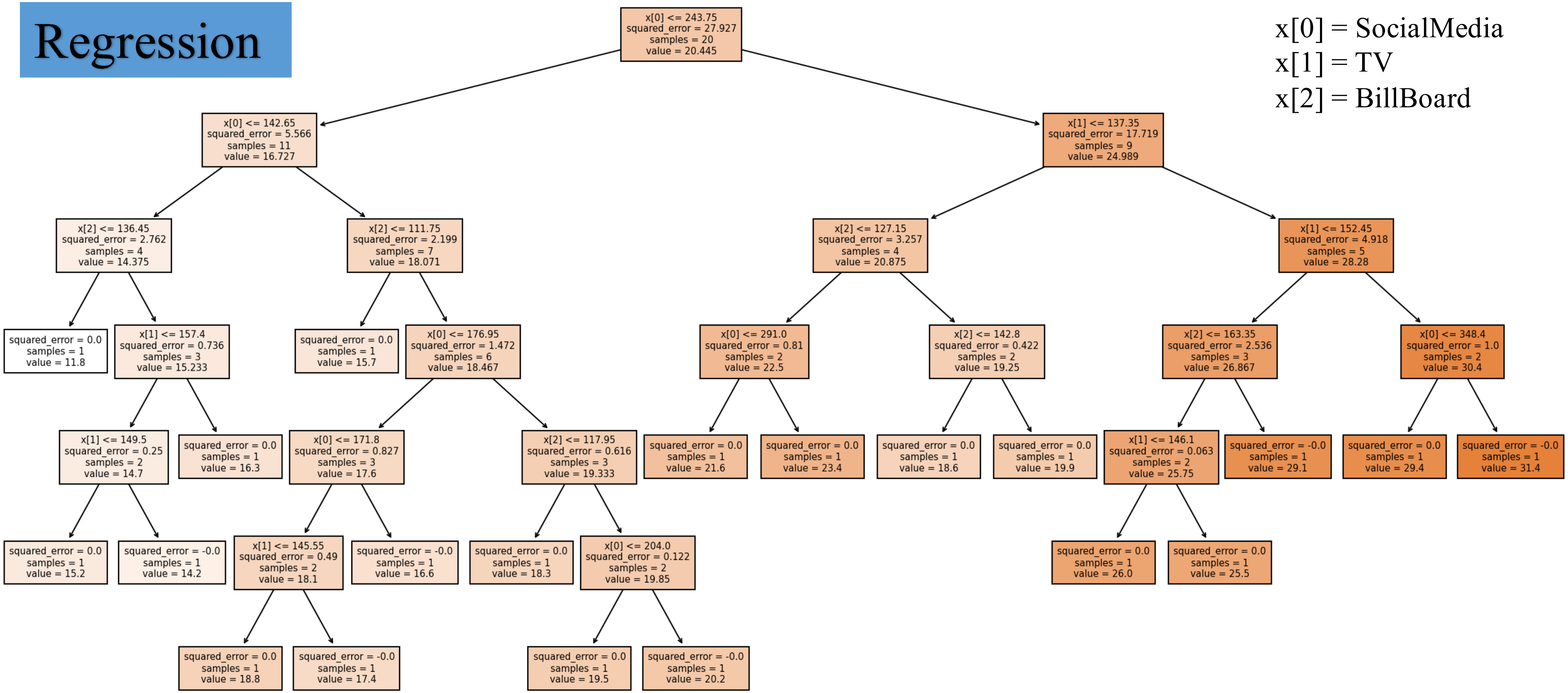
Default metric is MSE here

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(X, y_regression)
```

Decision Tree - Example

Regression

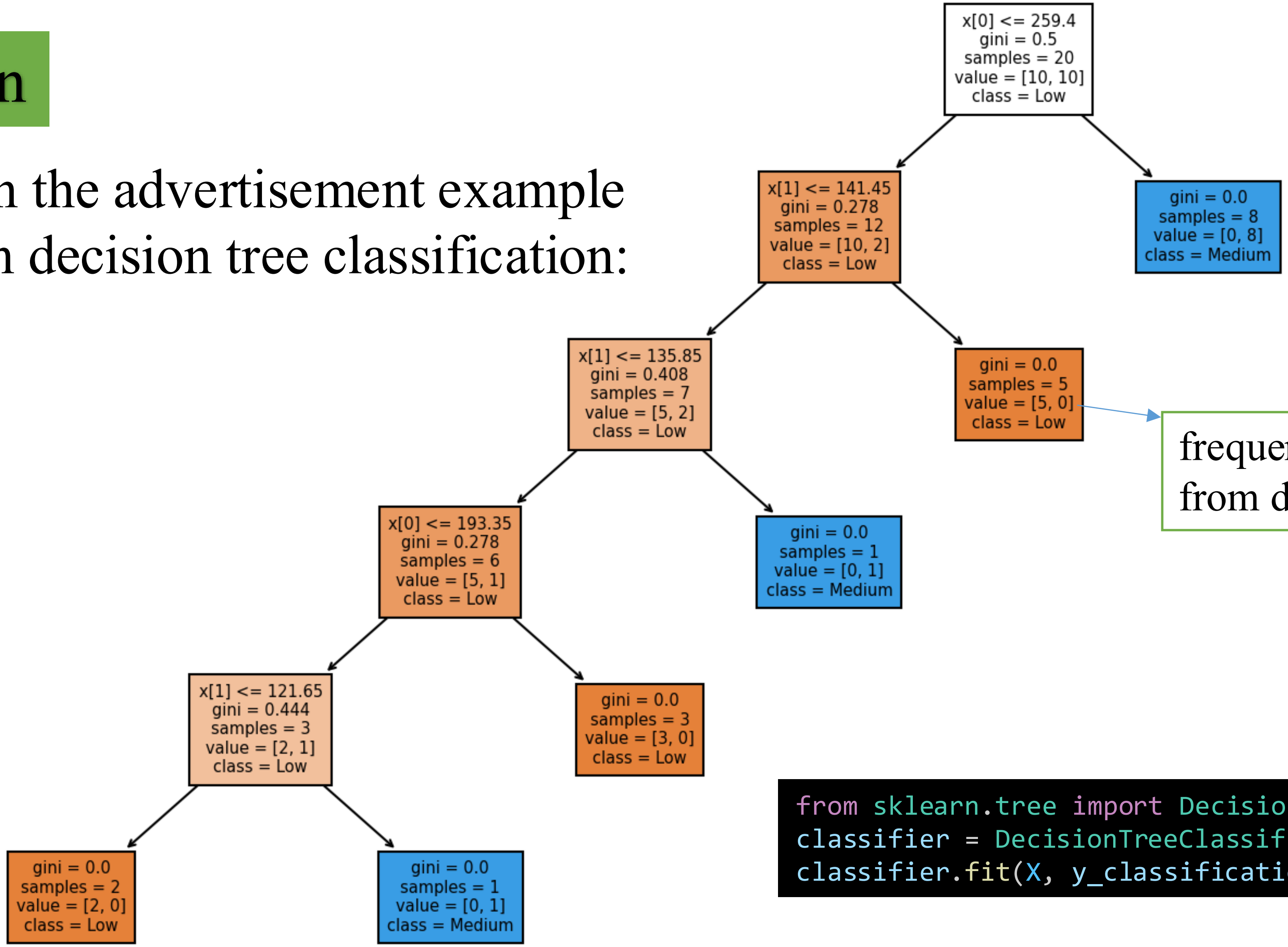
x[0] = SocialMedia
x[1] = TV
x[2] = BillBoard



Decision Tree - Example

Classification

✓ Based on the advertisement example if we run decision tree classification:



frequencies of instances from different classes

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion='gini')
classifier.fit(X, y_classification)
```

Decision Tree

Pruning

✓ Pruning, helps prevent overfitting in Decision Trees.

- **Pre-pruning**: Stop growing the tree early based on conditions like maximum depth, minimum samples per leaf, or minimum improvement.

- **Post-pruning**: Grow the full tree and then remove or merge nodes that don't significantly improve performance using validation data.

Decision Tree

Key advantages of Decision Trees

- ✓ Easy to understand and interpret, even for non-experts.
- ✓ Able to capture complex relationships and interactions between features.
- ✓ Handling both categorical and numerical features without requiring extensive preprocessing.
- ✓ Robust to outliers and missing values.

Limitations of Decision Trees

- ✓ Likely to overfit if we let the tree becomes too deep or complex.
- ✓ It is sensitive to small changes in the data, (leading to have different tree structures).
- ✓ It is Limited in handling continuous dependent variables (e.g. predicting the price of a house).

Decision Tree

Real-world applications of Decision Trees

- ✓ Disease diagnosis and medical decision-making.
- ✓ Customer segmentation.
- ✓ Fraud detection and anomaly detection.
- ✓ Recommender systems and personalized recommendations.
- ✓ Credit scoring and risk assessment.

Decision Tree

How Decision Trees contribute to advanced algorithms:

- ✓ Ensemble methods (combining multiple models instead of using a single model).
  - e.g., Random Forests, Gradient Boosting using it as base learners.
- ✓ Decision Trees can provide initial splits in hierarchical clustering algorithms.
- ✓ Can be used as feature selection and variable importance in model interpretability techniques.

Random Forest algorithm

Definition

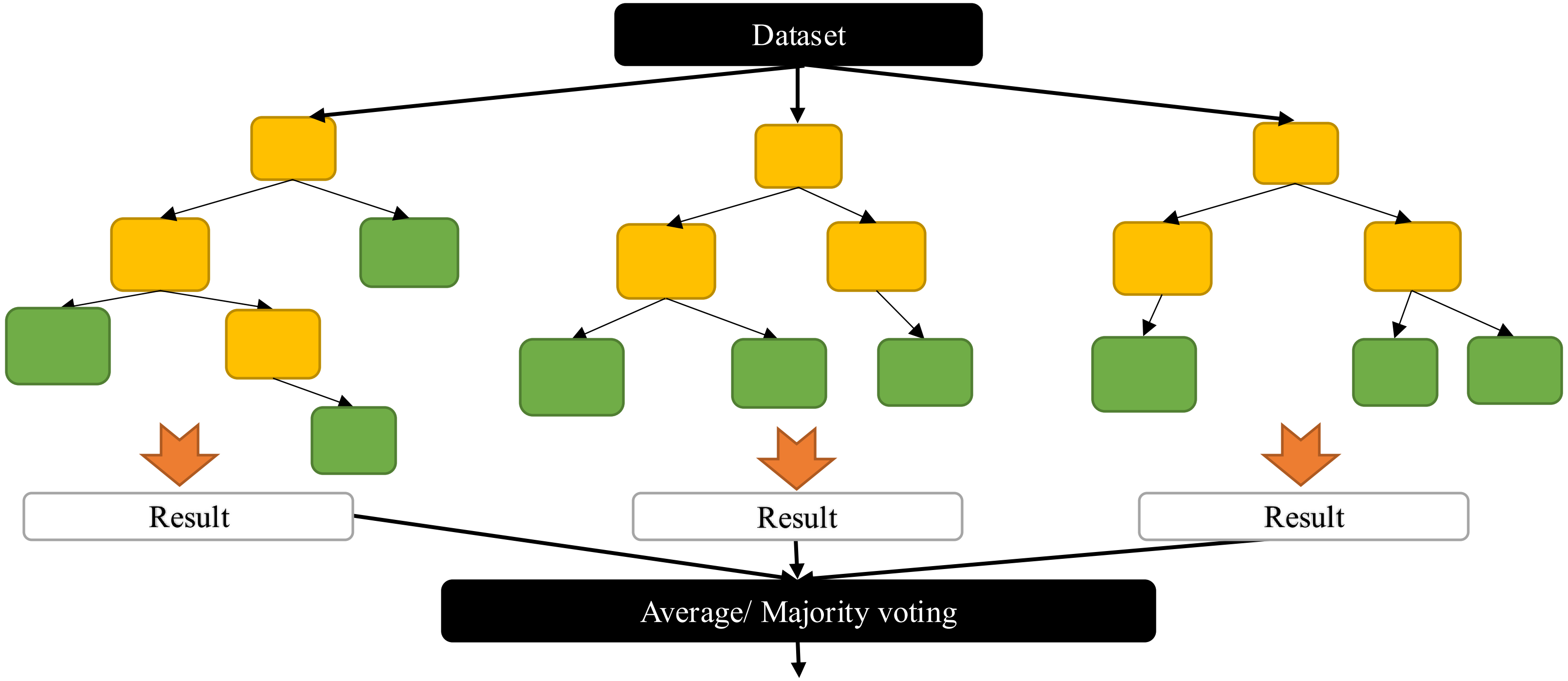
- ✓ Random Forest is an ensemble learning method .
- ✓ A common ML algorithm that combines multiple Decision Trees.
- ✓ To propose is to generate a single output.

Ensemble of Decision Trees

- ✓ Random Forest creates multiple decision trees.
- ✓ Each tree is trained on a different subset of the training data and randomly selected features.

Random Forest algorithm

Ensemble of Decision Trees



Random Forest algorithm

Randomness among trees

Random Sampling:

- ✓ During training, we select a random subset of the training data with allowing replacement (called bagging).

Random Feature Subset:

- ✓ At each split point in a decision tree, randomly selects a subset of features (independent variables) to consider for determining the best split.
- ✓ This randomness helps reduce overfitting.

Random Forest algorithm

Building Trees in Random Forest

✓ For each tree we use same recursive process as a regular decision tree algorithm.

In Prediction tasks:

- ✓ The final class label is determined by voting the predictions from all the decision trees.

In regression tasks:

- ✓ The final prediction is obtained by averaging the predictions from all trees.

Random Forest algorithm

Advantages

- ✓ Overfitting problem Reduction
- ✓ Robustness to outliers and noise
- ✓ Estimation of the feature importance at the same time

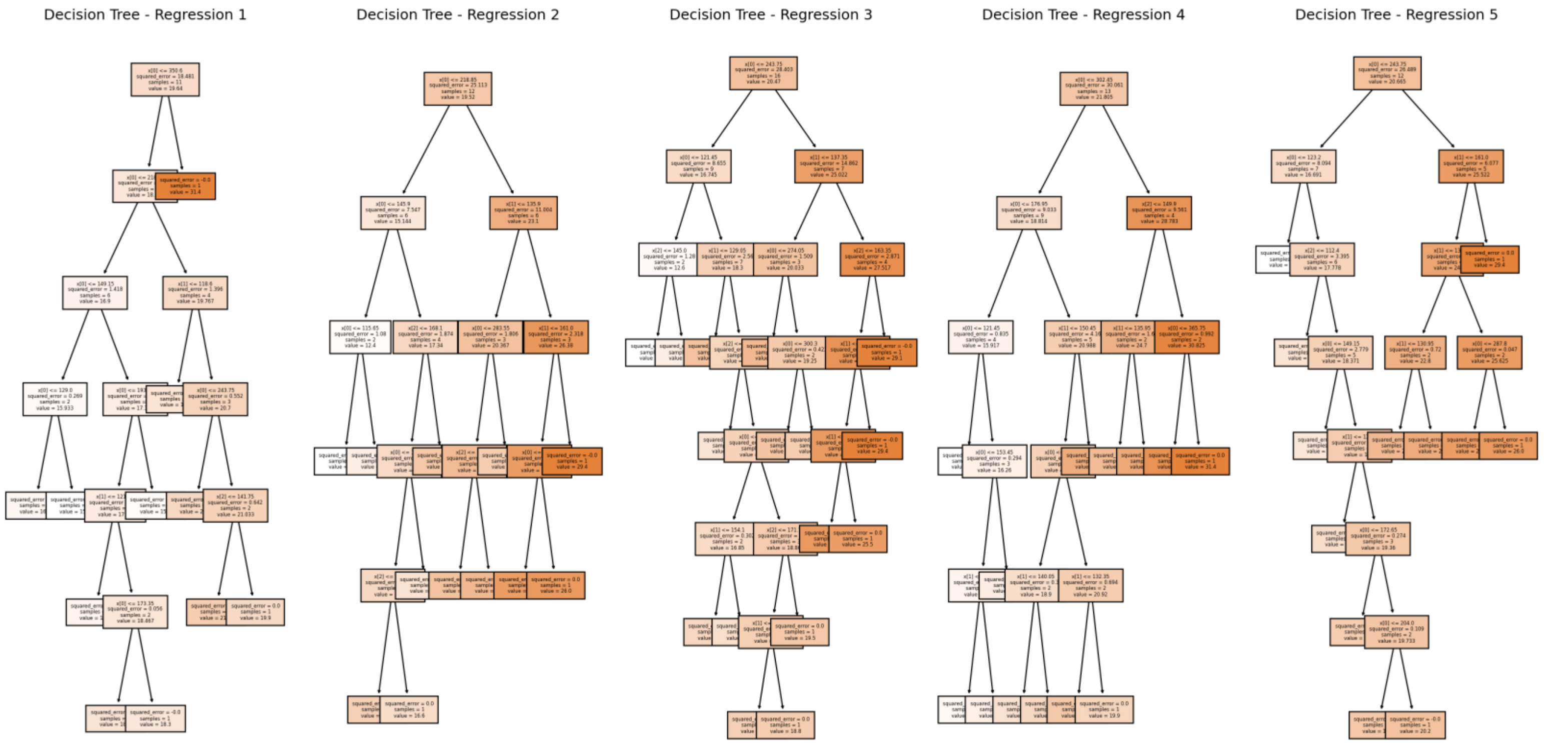
Disadvantages

- ✓ Memory and Computational Requirements
- ✓ Difficult of Interpretability
- ✓ Still can have overfitting

Random Forest algorithm

Regression

✓ Based on the advertisement example if we run decision tree regression:




```python
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators=5, random_state=10)
regressor.fit(X, y_regression)
```

Summery

- ✓ We answered what is the Decision Tree algorithm
- ✓ We introduced Different Splitting Measures for Decision Tree algorithm
- ✓ We discussed How to find thresholds and best feature
- ✓ We discussed advantages and disadvantages of Decision Tree algorithm
- ✓ We saw the example of Decision Tree algorithm for both classification and regression task
- ✓ We Understood Random Forest algorithm

Practice

Continue given example by 1) compare decision tree and random forest on given or desired dataset based on proper metrics. (classification or regression).