6.03 Random Forests

Ensemble Methods

- ML methods that combine several base models to produce one optimal predictive model
- Combine decisions from multiple models to improve the overall performance
- Ensemble learning involves creating a collection (or “ensemble”) of multiple algos for purpose of generating a single model that outperforms its base models

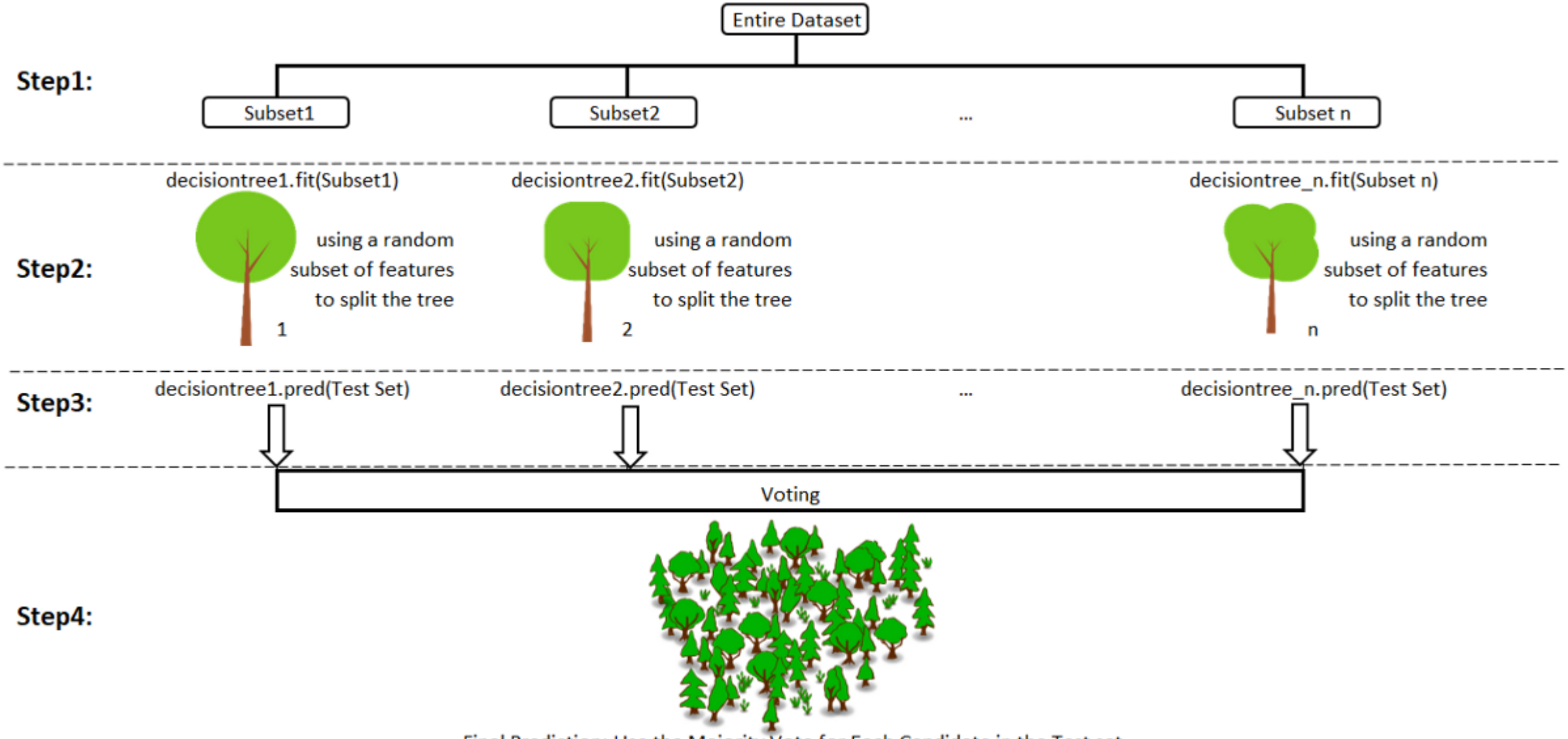
Random Forest (RF)

- Ensemble of decision trees using randomly sampled data
- Uses a variant of Bagging method where each time a split is considered only a portion of features are considered split candidates, so as to avoid the case with Bagging where very strong features can result in most trees using that feature as the top split
- Useful for both classification and regression

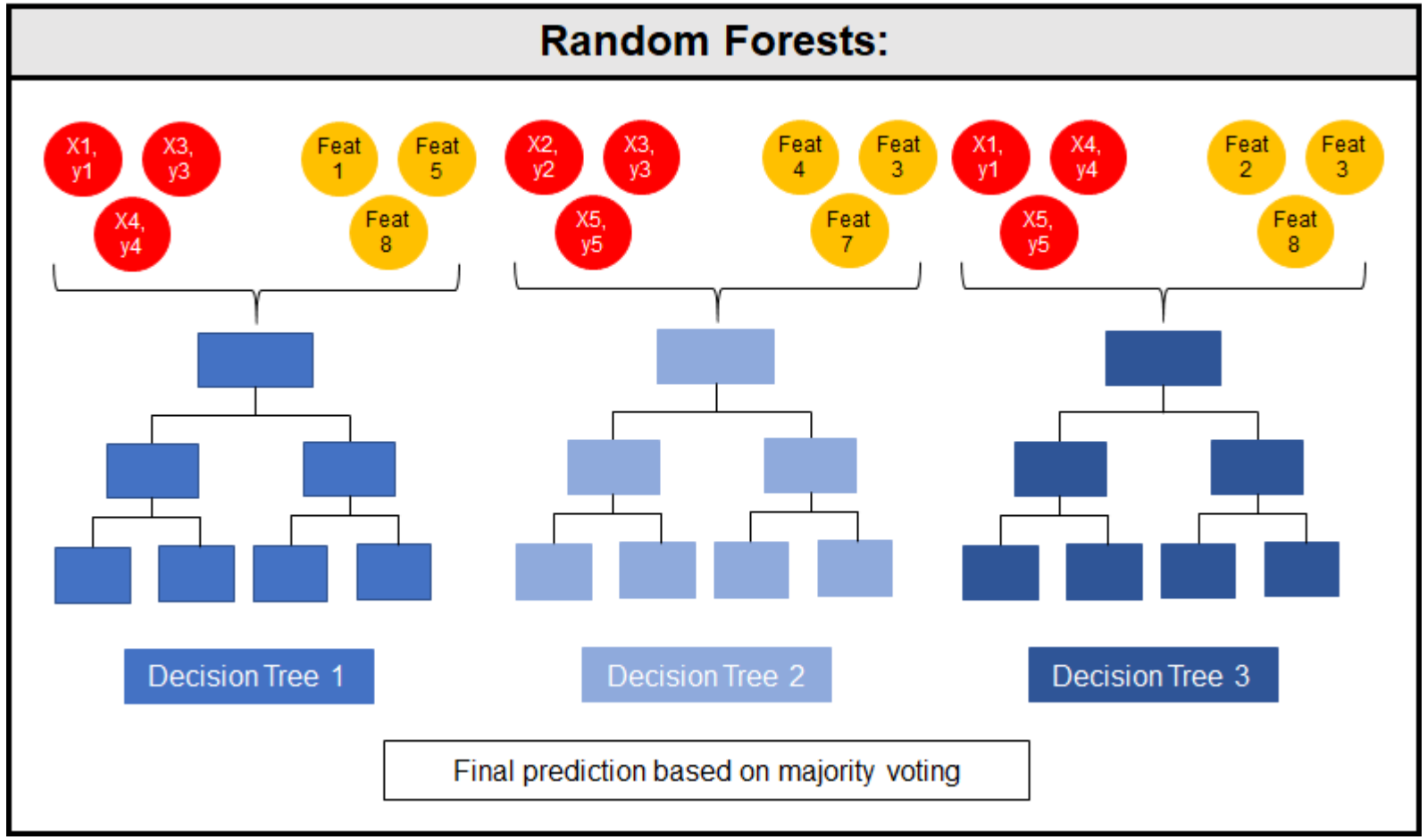
Random Forest

- When predicting a new value for a target feature, each tree is either using regression or classification to derive a value a.k.a. “vote”
- Random Forest algo then takes an average value or most popular category of all the trees in the ensemble
- This average is the predicted value of the target variable

Random Forest



Random Forest



Visualization of the random forests algorithm. Subsamples of the training samples and of the features are used to grow multiple weak learners (decision trees)

Random Forest Process

1. Create a random sample from the original data
2. Randomly select a set of features at each node in the decision tree
3. Decide the best split
4. For each data sample, create a separate base model
5. Compute the final prediction by average the predictions from all the individual models

Random Forests Pros

- Easy to understand
- Useful for data exploration
- Reduced data cleaning (scaling not required)
- Handle multiple data types
- Highly flexible and gives a good accuracy
- Works well on large datasets
- Overfitting is avoided (due to averaging)

Random Forests Cons

- Does not work well with sparse datasets
- Computationally expensive
- No interpretability