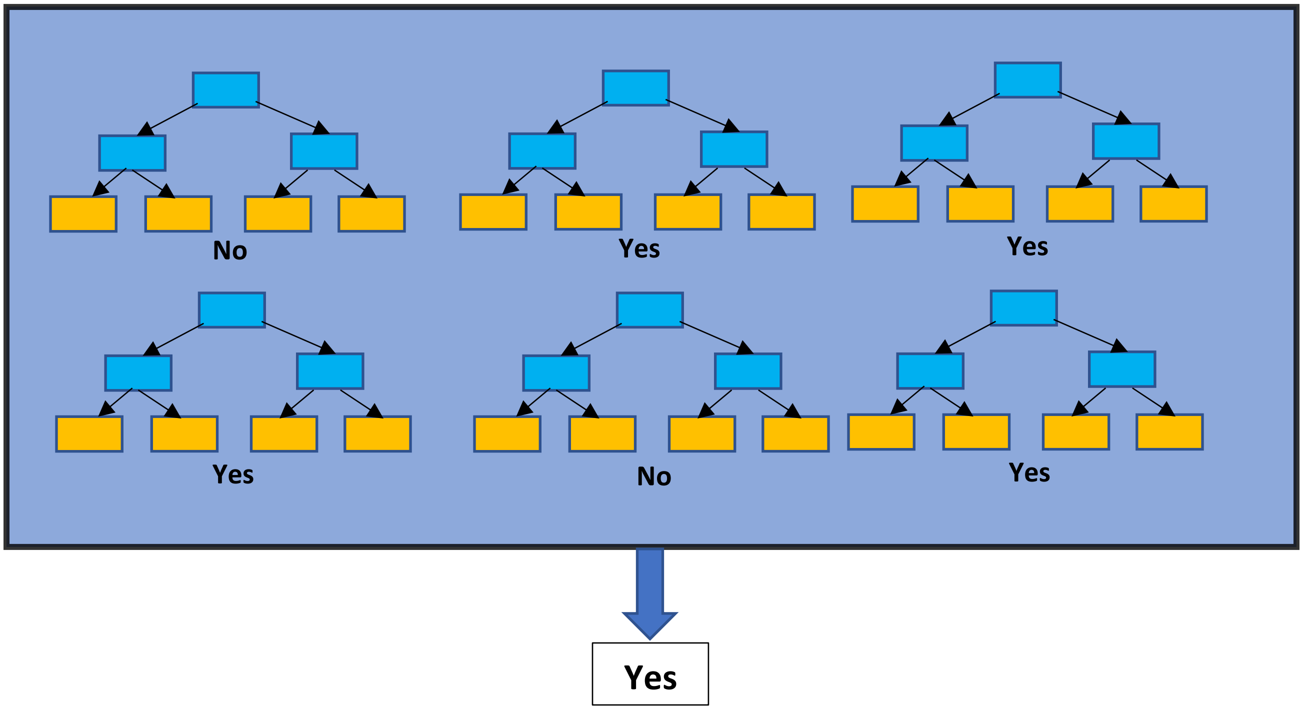
Random Forest Classifier

1. Bagging or Bootstrap Aggregating

- Train a set of decision trees given a dataset of n samples:
    - Repeat T times:
        - Step 1: Bootstrap: sample, **with replacement**, n training data points from the dataset.
        - Step 2: Build a decision tree for the set in Step 1.
        - Step 3: Save the prediction of the decision tree in Step 2 for the new data point
    - Prediction: Aggregate the predictions from the individual decision trees in Step 1, 2 and 3 and do one of the following:
        - **Majority Vote:** if the trees produce class labels (Categorical Data).
        - **Average:** if the trees produce numerical values (e.g. when predicting weight, price, etc.).



2. From Bagging to Random Forest (RF)

- Why RF:
  - No overfitting: Use multiple trees reduce the risk of overfishing
  - Training Time is less
  - High Accuracy: High accuracy for large datasets
  - Missing Data: RF can maintain accuracy when data is missing
- RF are made out of decision trees
- Bagged decision trees characteristics:
  - Have only the number of trees T in the previous algorithm.
  - Consider all the features of the dataset
- Random Forest (RF):
  - Same as Bagging but considers only a subset of features.
  - **Feature Bagging**: RF only try a subset of the features, usually of size

$$\frac{\sqrt{s}}{s} \quad \text{OR} \quad \frac{s}{3}$$

Where s is the number of features in the dataset

  - Reduce overfitting: This inject randomness that makes individual trees more unique
  - Improve overall performance

- **RF Algorithm:**

1. Create a bootstrapped dataset
   - Randomly select samples (data points) from the original dataset to create a bootstrapped dataset of the same size as the original dataset
   - Randomly selected samples can be repeated: Allow to pick a sample more than once (sample with replacement)
2. Build a Decision Tree

3. Repeat Step 1 & 2 several times (Typical number of trees is10, 30 or 100. Sometimes can be around 300 trees)

- Example:
  - Original Dataset:

Student	Income	Credit Rating	Buy Latest Smart Phone
Yes	High	Good	Yes
No	Low	Poor	No
Yes	High	Fair	Yes
Yes	Mid	Fair	Yes
No	Low	Fair	No
Yes	High	Good	No
Yes	Mid	Poor	No

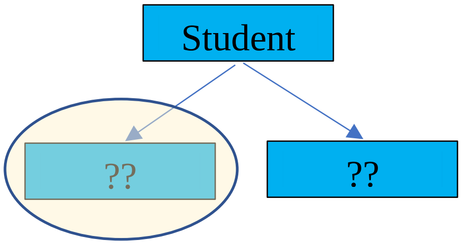
  - Step 1: Create a Bootstrapped Dataset: Random select data points for the original dataset

Student	Income	Credit Rating	Buy Latest Smart Phone
Yes	High	Good	Yes
Yes	Mid	Fair	Yes
Yes	Mid	Fair	Yes
No	Low	Poor	No
Yes	High	Good	Yes
No	Low	Poor	No
Yes	High	Good	Yes

  - Step 2: Create a decision tree using the bootstrapped dataset:
    - Use a subset of features at each step: Pick for example: **Student** & **Credit Rating**.
    - Assume **Student** has better value to split the Root (higher entropy or low Gini index):

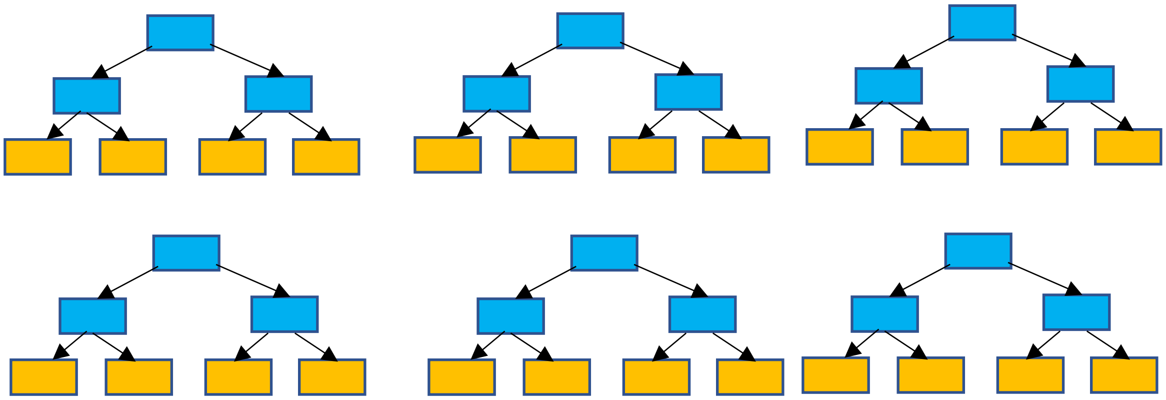
Student	Income	Credit Rating	Buy Latest Smart Phone
Yes	High	Good	Yes
Yes	Mid	Fair	Yes
Yes	Mid	Fair	Yes
No	Low	Poor	No
Yes	High	Good	Yes
No	Low	Poor	No
Yes	High	Good	Yes

- Split the data based on the Student
- Remove Student for next choices.



- Now we need to split data points in each node:
  - Let us consider two random features: For example **Income & Credit Rating**
- Continue building the tree by considering a random set of features at each node to split the data points

- Step 3: Repeat Step 1 and 2 for a new Bootstrapped dataset



- **Out-Of-Bag Dataset:**
  - Typically not all the data points are included in generating a bootstrap dataset: Typically 1/3 of dataset is not included in the bootstrapped dataset.
  - Data points that were not used in generating the decision trees for the Out-Of-Bag dataset
    - Original Dataset

Student	Income	Credit Rating	Buy Latest Smart Phone
Yes	High	Good	Yes
No	Low	Poor	No
Yes	High	Fair	Yes
Yes	Mid	Fair	Yes
No	Low	Fair	No
Yes	High	Good	No
Yes	Mid	Poor	No

    - Out-Of-Bag dataset

Student	Income	Credit Rating	Buy Latest Smart Phone
Yes	Mid	Fair	Yes
No	Low	Fair	No
Yes	High	Good	No
Yes	Mid	Poor	No

- Use the data points in Out-Of-Bag dataset to measure the accuracy of your RF:
    - Step 1: Consider the first Bootstrap Dataset:
        - Pick a sample from the Out-Of-Bag dataset and run it through all the decision tree in the RF generated.
        - Label the sample with the label that has a majority vote.
    - Step 2: Repeat Step 1
- Accuracy of the RF:
    - It is measured by the proportion of the Out-Of-Bag samples that were correctly classified.
    - The proportion of the Out-Of-Bag samples that were not correctly classified is “Out-Of-Bag Error”.

- **The Buck Does not Stop Here:**
    - In the previous example, we only considered two features at each note
    - Run the RF with three features, 4 features, etc.
    - Compare the Out-Of-Bag Error for each run
    - Choose the RF that the smallest Out-Of-Bag Error.