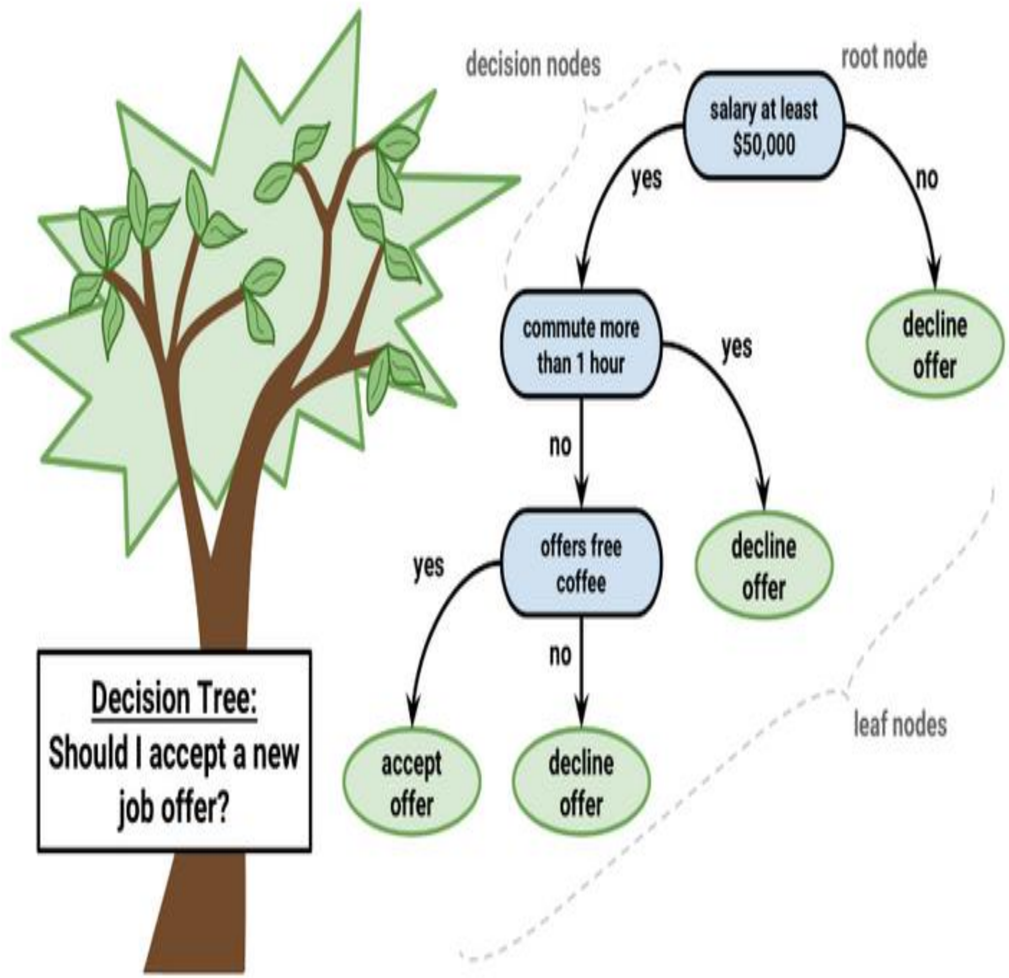
# INT354
# Machine Learning

## Decision Trees and Random Forests

# Decision Tree Learning

- **Breaking down data by making decisions based on asking a series of questions.**

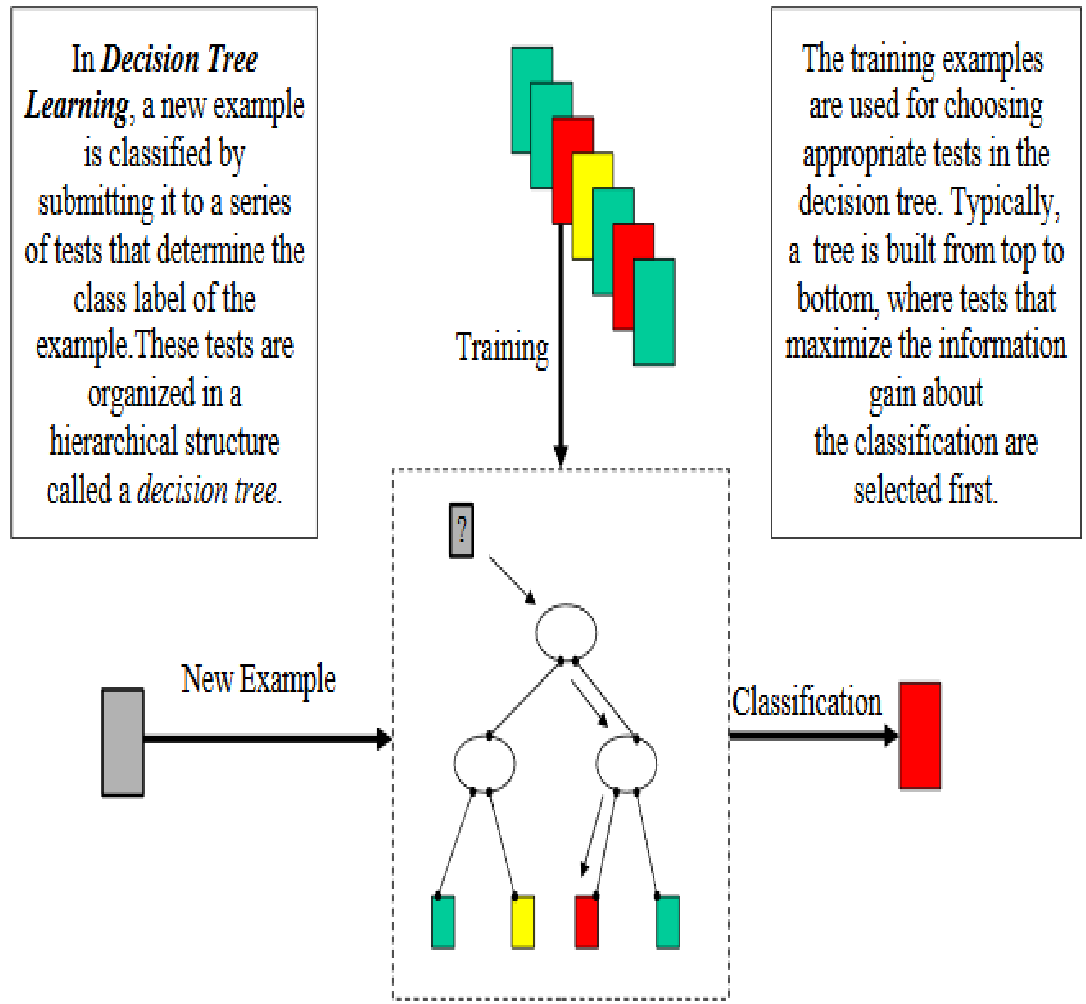


# Decision Tree Learning

- **A decision tree consists of**
  - **Nodes**
    - **Test for the value of a certain attribute.**
  - **Edges**
    - **Correspond to the outcome of a test.**
    - **Connect to the next node or leaf.**
  - **Leaves**
    - **Terminal nodes that predict the outcome.**

# Algorithm

1. **Start at the root.**
2. **Perform the test.**
3. **Follow the edge corresponding to outcome.**
4. **Goto step 2 unless leaf.**
5. **Predict the outcome associated with the leaf.**



# Design Decision Tree

| Day | Temperature | Outlook | Humidity | Windy | Play Golf? |
|---|---|---|---|---|---|
| 07-05 | hot | sunny | high | false | no |
| 07-06 | hot | sunny | high | true | no |
| 07-07 | hot | overcast | high | false | yes |
| 07-09 | cool | rain | normal | false | yes |
| 07-10 | cool | overcast | normal | true | yes |
| 07-12 | mild | sunny | high | false | no |
| 07-14 | cool | sunny | normal | false | yes |
| 07-15 | mild | rain | normal | false | yes |
| 07-20 | mild | sunny | normal | true | yes |
| 07-21 | mild | overcast | high | true | yes |
| 07-22 | hot | overcast | normal | false | yes |
| 07-23 | mild | rain | high | true | no |
| 07-26 | cool | rain | normal | true | no |
| 07-30 | mild | rain | high | false | yes |

**Note: Consider "Outlook" as root node**

# Exercise

| | | | | | |
|---|---|---|---|---|---|
| today | cool | sunny | normal | false | ? |

| | | | | | |
|---|---|---|---|---|---|
| tomorrow | mild | sunny | normal | false | ? |

# Good Attribute?

- **Prefers attributes that have a high degree of "order".**
  - **Maximum Order: All examples are of the same class.**
  - **Minimum Order: All classes are equally likely.**

# Entropy

**Entropy is the measure of**

- **Impurity**
- **Disorder**
- **uncertainty**

**Entropy for two classes:**

$$\boldsymbol{E(S) = -p_1 . log_2 p_1 - p_2 . log_2 p_2}$$

**S is the complete set,**
**p1 is the proportion of examples in class 1 and**
**p2=1-p1 is the proportion of examples in class 2.**

# Exercise

**Compute entropy**

- **Outlook=sunny**
- **Outlook=overcast**
- **Outlook=rainy**

# Entropy for more classes

$$E(S) = -p_1.logp_1 - p_2.logp_2 \dots - pn.logpn$$

$$= -\sum_{i=1}^{n} p_i.logpi$$

# Entropy Disadvantages

- **Only computes the quality of a single subset of examples corresponds to a single value.**

# Average Entropy

- **Computes the quality of the entire split corresponds to an entire attribute.**

$$I(S,A) = \sum_{i} \frac{|s_i|}{|s|} . E(Si)$$

**S is the complete set.**

**A is Attribute that splits S into different subsets.**

# Properties of Entropy

- **When node is pure, measure should be zero.**
- **When impurity is maximal, measure should be maximal.**

# Exercise

- **Compute the average entropy for attribute "Outlook".**

# Information Gain

$$Gain(S,A) = E(S) - I(S,A)$$
$$= E(S) - \sum_i \frac{|s_i|}{|s|} . E(Si)$$

**Where *S* is complete set and *A* is the attribute that splits the set *S* into subsets $S_i$.**

**Note: Maximizing information gain is equivalent to minimizing average entropy, because *E(S)* is constant for all attributes *A*.**

# Exercise

- **Compute the information gain for**
  - **Gain(S, Humidity)=?**
  - **Gain(S, Wind)=?**
  - **Gain(S, Outlook)=?**
  - **Gain(S, Temperature)=?**

**Note:**

**The attribute with maximum information gain is selected as root node.**

# Intrinsic Information of an attribute

- **Amount of information need to tell which branch an instance belongs to**

$$Int1(S,A) = -\sum_{i} \frac{|S_i|}{|S|} log(\frac{|S_i|}{|S|})$$

**Note:**

**Attributes with higher intrinsic information are less useful.**

# Exercise

- Compute intrinsic information of
  - Day
  - Temperature
  - Humidity
  - Windy
  - Outlook

# Gain Ratio

- **Modification of the information gain that reduces its bias towards multi-valued attributes.**

$$GR(S, A) = \frac{Gain(S, A)}{Int1(S, A)}$$

# Exercise

- **Compute gain ratio of**
  - **Day**
  - **Temperature**
  - **Humidity**
  - **Windy**
  - **outlook**

# Gini Index

- **Measure impurity**

$$Gini(S) = 1 - \sum_i p_i^{\,2}$$

- **Average Gini Index**

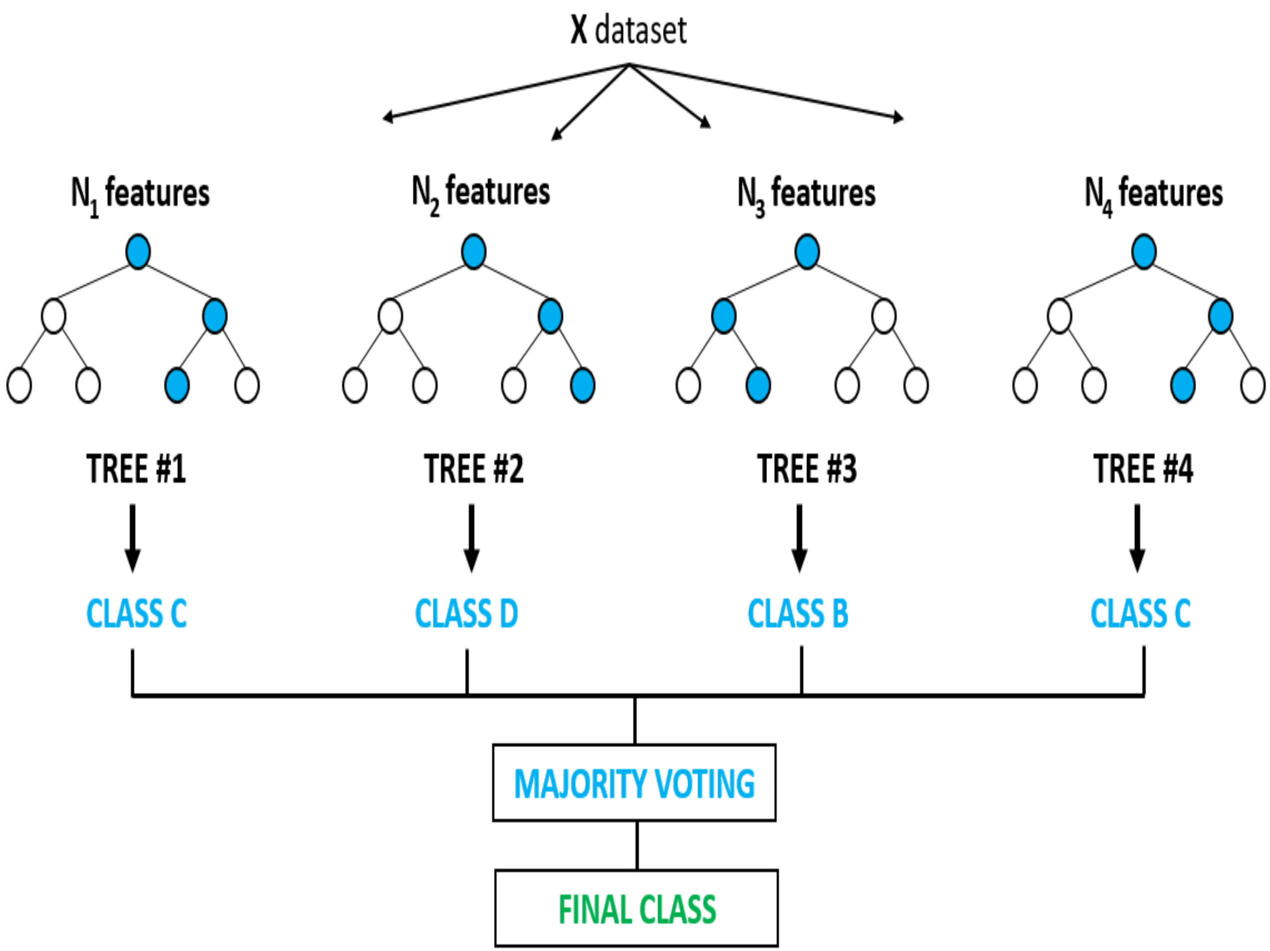
$$Gini(S, A) = \sum_i \frac{|S_i|}{|S|} . Gini(Si)$$

# Pruning

- **Pre-pruning:**
  - **Stop growing a branch when information becomes unreliable.**
- **Post-pruning:**
  - **Grow a decision tree that correctly classifies all training data.**
  - **Simplify it later by replacing some nodes with leafs.**

# Random Forest

- **Builds multiple decision tree and merges them together to get a more accurate and stable prediction.**



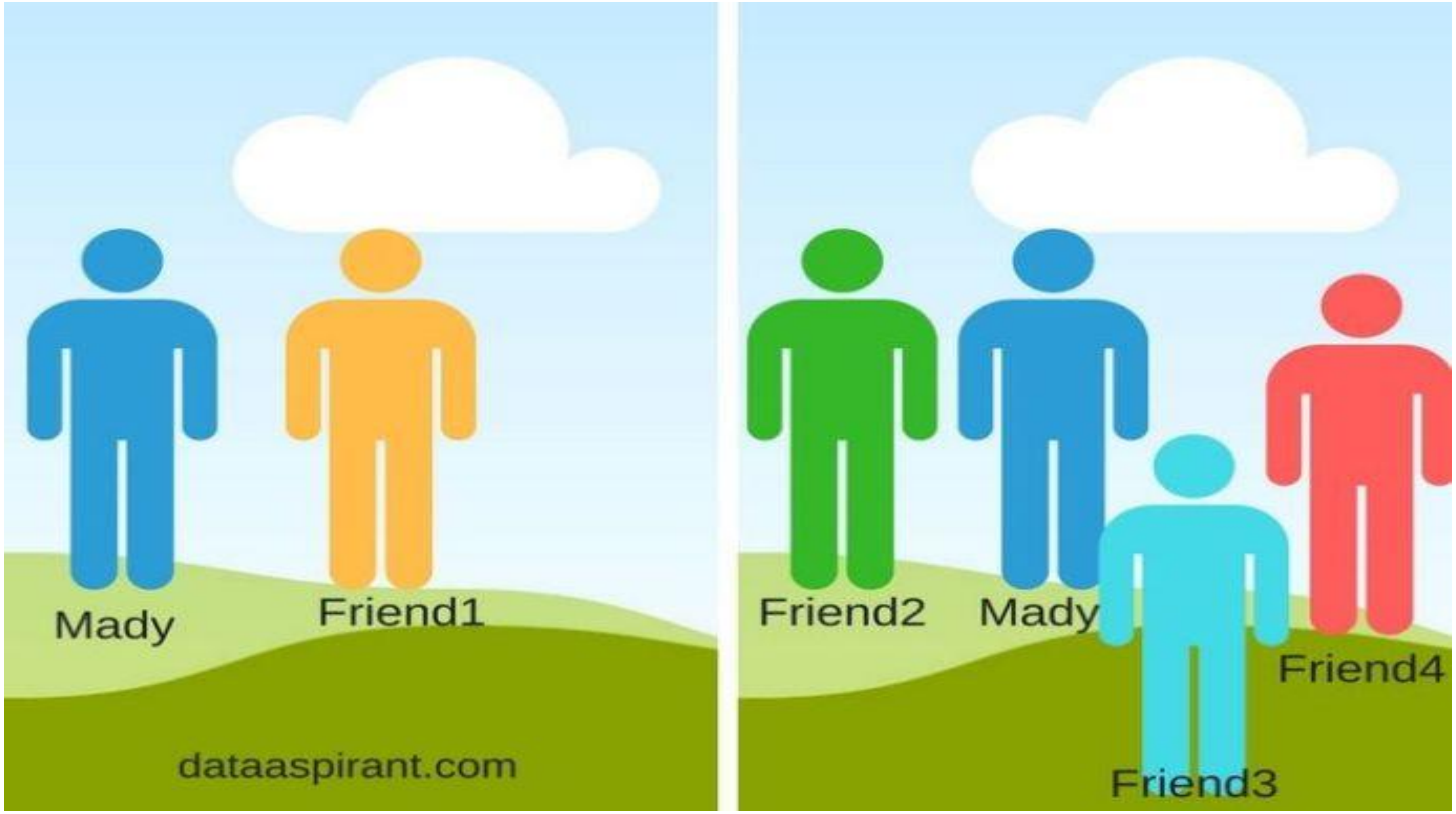
# Random Forest Creation Algorithm

1. Randomly select “K” features from total “m” feaures.
   - Where K << m
2. Among the “K” features, calculate the node “d” using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until “l” number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for “n” number of times to create “n” number of trees.

# Random Forest Prediction Algorithm

1. Takes the **test features** and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome.
2. Calculate the **vote** for each predicted target.
3. Consider the **high voted predicted target** as the final prediction from the random forest algorithm.

# Random Forest Example

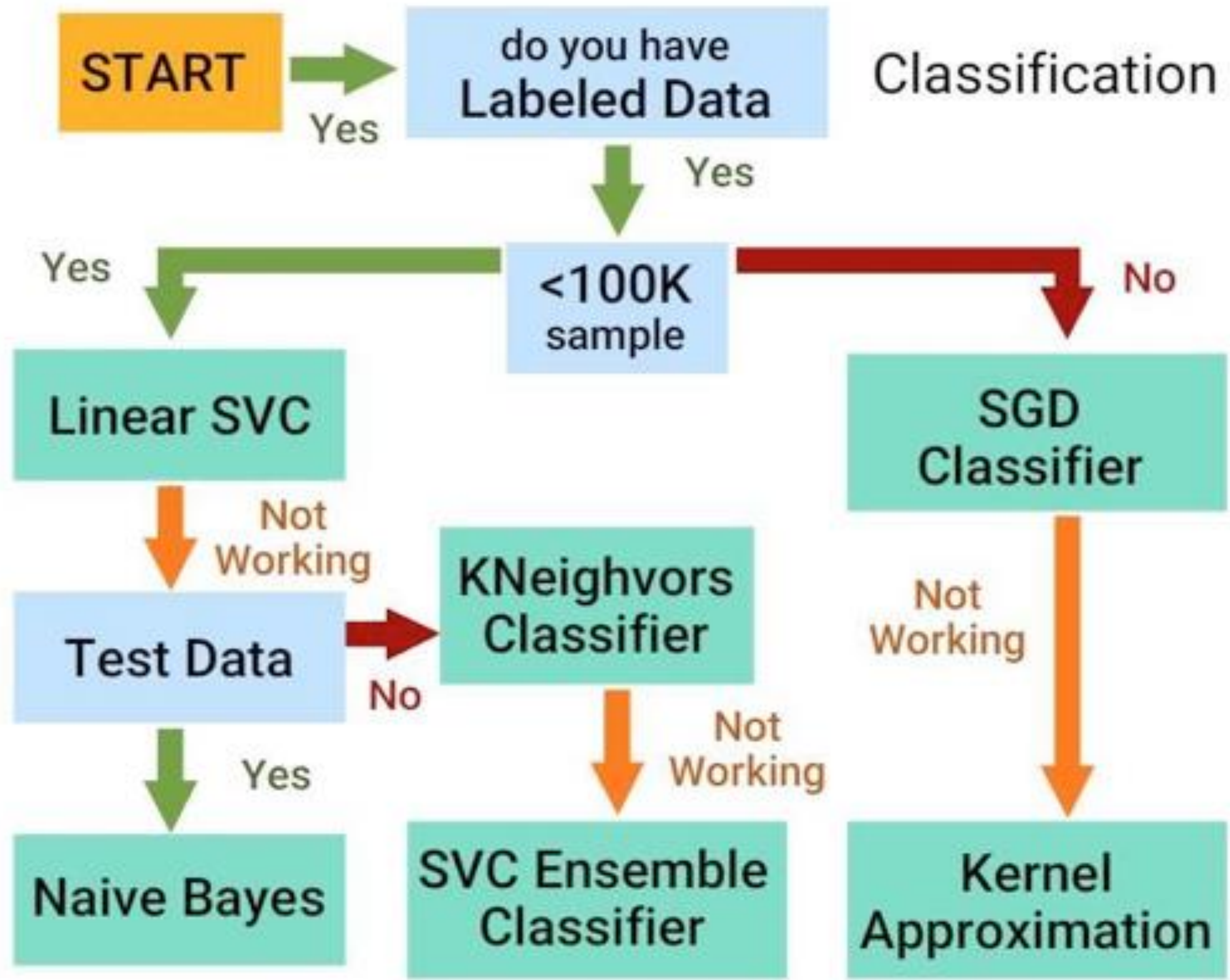



- **Mady takes opinion from friends for some problem.**

# Why Random Forest Algorithm?

- **Can be used for both classification and regression.**
- **Handles the missing value.**
- **Not over-fit the model.**

# Algorithm Selection




COMING UP