

Embedded Methods

Feature Importance from

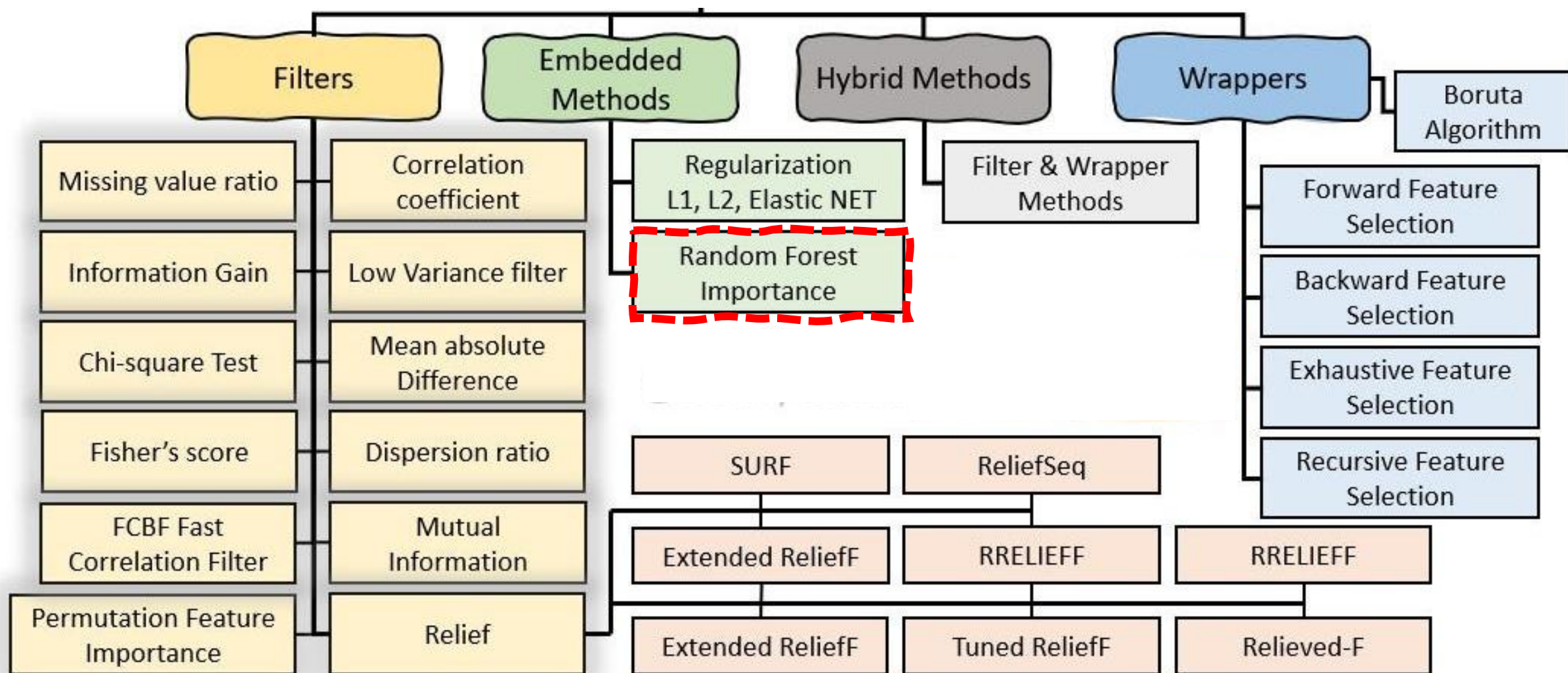
Random Forest

Nov 14 2025

Giảng viên: TS. Lưu Phúc Lợi

Luu.p.loi@gmail.com

Zalo: 0901802182



Content

- What is cross entropy and conditional entropy?
- What is Information Gain?
- Decision Tree
- Bagging
- Boosting
- Random Forest
- Feature Importance from Random Forest
- Hands-on Exercises

Parametric Vs Non-parametric Classification Models

We can divide classification models into two broad categories.

- Parametric Models (Where we are trying to optimize for some parameter)
 - Logistic Regression (Optimize for θ)

$$\min_{\theta} \underbrace{\frac{1}{n} \sum_{i=1}^n \left[p_i(e_1) \log \left(\frac{1}{1 + e^{-x_i^\top \theta}} \right) + p_i(e_2) \log \left(\frac{1}{1 - \frac{1}{1 + e^{-x_i^\top \theta}}} \right) \right]}_{\text{objective function as } \mathcal{L}} \quad \text{where } \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

- SVM (Optimize for w)

$$\min_w \quad ||w||_2^2 - \sum_{i=1}^n y_i (w^\top x_i - b).$$

- Non-parametric Models (There isn't a set of parameter to optimize)
 - K Nearest Neighbor (Finding k nearest samples and set the label to them)
 - **Decision Tree** (This is what we are going to learn today)

Entropy and Conditional Entropy

To understand decision trees, you must first know 2 concepts really well.

- Entropy

$$H(X) = \mathbb{E} \left[\log \left(\frac{1}{p(X)} \right) \right] = \sum_{x \in \mathcal{X}} \log \left(\frac{1}{p(X)} \right) p(x) \quad (2)$$

- Entropy is the **average** information for all the possible outcomes.

- Conditional Entropy

$$H(Y|X) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log(p(y|x)) \quad (3)$$

- Conditional Entropy is the **average** information for all the possible outcomes **after** you were told $X = x$ happened.
- Using Bayes Theorem where $p(x, y) = p(y|x)p(x)$, we can rewrite the conditional entropy equation as

$$H(Y|X) = - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(y|x)p(x) \log(p(y|x)) \quad (4)$$

$$= - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log(p(y|x)). \quad (5)$$

Eq. (4) is going to be the version we use for conditional entropy.

Quick Review of Entropy

- Calculate the entropy for these outcomes
- Let the random variable of play golf be Y
- Use log base 2.

$$H(X) = \mathbb{E} \left[\log \left(\frac{1}{p(X)} \right) \right] = \sum_{x \in \mathcal{X}} \log \left(\frac{1}{p(X)} \right) p(x)$$

Play Golf(14)	
Yes	No
9	5

Quick Review of Entropy

- Calculate the entropy for these outcomes
- Use log base 2.

$$H(X) = \mathbb{E} \left[\log \left(\frac{1}{p(X)} \right) \right] = \sum_{x \in \mathcal{X}} \log \left(\frac{1}{p(X)} \right) p(x)$$

$$H(Y) = -\frac{9}{14} \log \left(\frac{9}{14} \right) - \frac{5}{14} \log \left(\frac{5}{14} \right) \approx 0.94.$$

Play Golf(14)	
Yes	No
9	5

Practice calculating Conditional Entropy

- Calculate the conditional entropy for these outcomes
- Let x be the Outlook
- Let y be the Play golf decision
- Use log base 2.

		PlayGolf(14)		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5

$$\begin{aligned} H(Y|X) &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(y|x)p(x) \log(p(y|x)) \\ &= - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log(p(y|x)). \end{aligned}$$

$$\begin{aligned}
 H(Y|X) &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(y|x)p(x) \log(p(y|x)) \\
 &= - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log(p(y|x)).
 \end{aligned}$$

		PlayGolf(14)		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5

Example Calculating Entropy and Conditional Entropy

$$\begin{aligned}
 H(Y|X) &= -p(x = \text{sunny}) [p(y = \text{yes}|x = \text{sunny}) \log_2(p(y = \text{yes}|x = \text{sunny})) + p(y = \text{no}|x = \text{sunny}) \log_2(p(y = \text{no}|x = \text{sunny}))] \\
 &\quad - p(x = \text{overcast}) [p(y = \text{yes}|x = \text{overcast}) \log_2(p(y = \text{yes}|x = \text{overcast})) + p(y = \text{no}|x = \text{overcast}) \log_2(p(y = \text{no}|x = \text{overcast}))] \\
 &\quad - p(x = \text{Rainy}) [p(y = \text{yes}|x = \text{Rainy}) \log_2(p(y = \text{yes}|x = \text{Rainy})) + p(y = \text{no}|x = \text{Rainy}) \log_2(p(y = \text{no}|x = \text{Rainy}))]
 \end{aligned}$$

$$\begin{aligned}
 H(Y|X) &= -\frac{5}{14} \left[\frac{3}{5} \log_2\left(\frac{3}{5}\right) + \frac{2}{5} \log_2\left(\frac{2}{5}\right) \right] \\
 &\quad - \frac{4}{14} [1 \log_2(1) + 0 \log_2(0)] \\
 &\quad - \frac{5}{14} \left[\frac{2}{5} \log_2\left(\frac{2}{5}\right) + \frac{3}{5} \log_2\left(\frac{3}{5}\right) \right] = 0.347 - 0 + 0.347 = 0.69
 \end{aligned}$$

Practice calculating Conditional Entropy

- Calculate the conditional entropy for these outcomes
- Let z be the Temperature
- Let y be the Play golf decision
- Use log base 2.

		PlayGolf(14)		
		Yes	No	
Temperature	Hot	2	2	4
	Cold	3	1	4
	Mild	4	2	6

$$\begin{aligned} H(Y|X) &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(y|x)p(x) \log(p(y|x)) \\ &= - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log(p(y|x)). \end{aligned}$$

$$\begin{aligned}
 H(Y|X) &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(y|x)p(x) \log(p(y|x)) \\
 &= - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log(p(y|x))
 \end{aligned}$$

		PlayGolf(14)		
		Yes	No	
Temperature	Hot	2	2	4
	Cold	3	1	4
	Mild	4	2	6

Example Calculating Entropy and Conditional Entropy

$$\begin{aligned}
 H(Y|Z) &= -p(z = \text{hot}) [p(y = \text{yes}|z = \text{hot}) \log_2(p(y = \text{yes}|z = \text{hot})) + p(y = \text{no}|z = \text{hot}) \log_2(p(y = \text{no}|z = \text{hot}))] \\
 &\quad - p(x = \text{cold}) [p(y = \text{yes}|z = \text{cold}) \log_2(p(y = \text{yes}|z = \text{cold})) + p(y = \text{no}|z = \text{cold}) \log_2(p(y = \text{no}|z = \text{cold}))] \\
 &\quad - p(x = \text{mild}) [p(y = \text{yes}|z = \text{mild}) \log_2(p(y = \text{yes}|z = \text{mild})) + p(y = \text{no}|z = \text{mild}) \log_2(p(y = \text{no}|z = \text{mild}))]
 \end{aligned}$$

$$\begin{aligned}
 H(Y|Z) &= -\frac{4}{14} \left[\frac{1}{2} \log_2\left(\frac{1}{2}\right) + \frac{1}{2} \log_2\left(\frac{1}{2}\right) \right] \\
 &\quad - \frac{4}{14} \left[\frac{3}{4} \log_2\left(\frac{3}{4}\right) + \frac{1}{4} \log_2\left(\frac{1}{4}\right) \right] \\
 &\quad - \frac{6}{14} \left[\frac{4}{6} \log_2\left(\frac{4}{6}\right) + \frac{2}{6} \log_2\left(\frac{2}{6}\right) \right] = 0.286 + 0.232 + 0.391 = 0.91
 \end{aligned}$$

Using Entropy and Conditional Entropy together.

Entropy $H(Y)$

- Is the **average** amount of information for an event of multiple possible outcomes.
- Entropy is based on chaos. The more random and chaotic, the more information exists in the system.

Cross Entropy $H(p, q)$

- Given the original randomness of p , Cross Entropy is the additional chaos added when distribution q is introduced.

Conditional Entropy $H(Y|X)$

- after X is revealed, there's less chaos and less information in the system Y .
- $H(Y|X)$ is the leftover information after X is revealed.
- It is always going to be equal to or less than the original amount of information,

$$H(Y|X) \leq H(Y). \quad (1)$$

The information gained by knowing X is the difference between the original $H(Y)$ and the conditional $H(Y|X)$

$$\text{Information Gained From } X = \Delta H_{Y|X=x} = H(Y) - H(Y|X = x). \quad (2)$$

- The bigger the $\Delta H_{Y|X=x}$, the more information X give us about Y .
- In general, we call the difference between original and after knowing X as **Information Gain**.

Which factor gives us more information on golf playing?

We now have 3 random variables X , Y , Z .

- $H(Y)$: Is the entropy of playing golf without any additional information
- $H(Y|X)$: Is the entropy of playing golf given weather outlook.
- $H(Y|Z)$: Is the entropy of playing golf given temperature.

Which factor gives us more information on golf playing?

$$H(Y) = -\frac{9}{14} \log\left(\frac{9}{14}\right) - \frac{5}{14} \log\left(\frac{5}{14}\right) \approx 0.94.$$

$$\begin{aligned} H(Y|X) &= -\frac{5}{14} \left[\frac{3}{5} \log_2\left(\frac{3}{5}\right) + \frac{2}{5} \log_2\left(\frac{2}{5}\right) \right] \\ &\quad - \frac{4}{14} [1 \log_2(1) + 0 \log_2(0)] \\ &\quad - \frac{5}{14} \left[\frac{2}{5} \log_2\left(\frac{2}{5}\right) + \frac{3}{5} \log_2\left(\frac{3}{5}\right) \right] = 0.347 - 0 + 0.347 = 0.69 \end{aligned}$$

$$\begin{aligned} H(Y|Z) &= -\frac{4}{14} \left[\frac{1}{2} \log_2\left(\frac{1}{2}\right) + \frac{1}{2} \log_2\left(\frac{1}{2}\right) \right] \\ &\quad - \frac{4}{14} \left[\frac{3}{4} \log_2\left(\frac{3}{4}\right) + \frac{1}{4} \log_2\left(\frac{1}{4}\right) \right] \\ &\quad - \frac{6}{14} \left[\frac{4}{6} \log_2\left(\frac{4}{6}\right) + \frac{2}{6} \log_2\left(\frac{2}{6}\right) \right] = 0.286 + 0.232 + 0.391 = 0.91 \end{aligned}$$

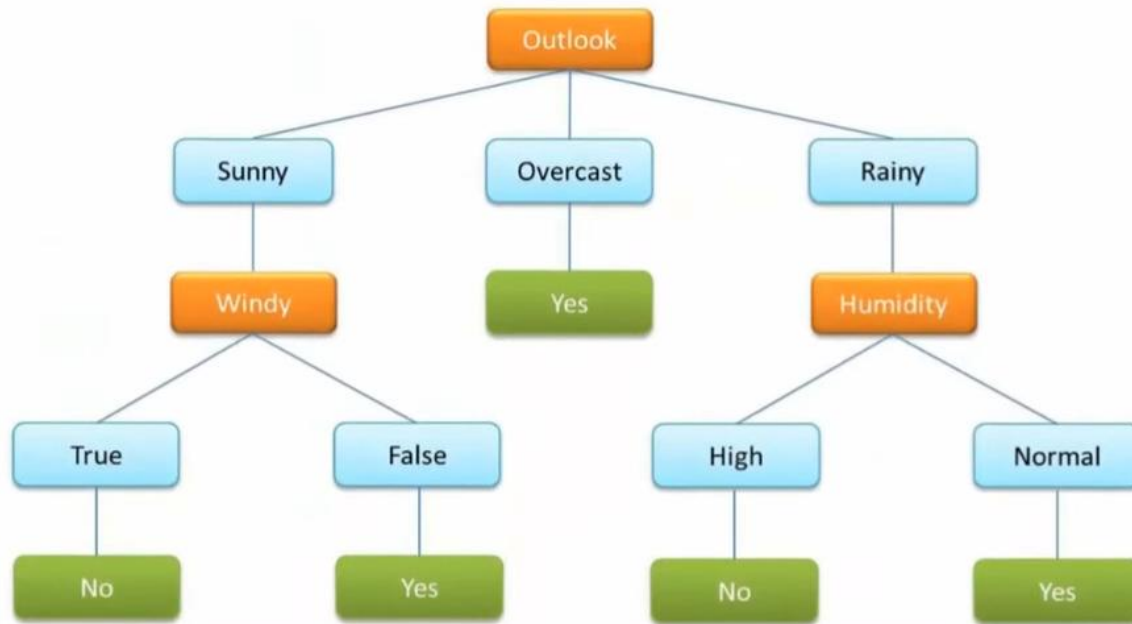
Play Golf(14)	
Yes	No
9	5

		PlayGolf(14)		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5

		PlayGolf(14)		
		Yes	No	
Temperature	Hot	2	2	4
	Cold	3	1	4
	Mild	4	2	6

We are now ready to talk about the Decision Tree Classifier

- Given past historical data of golf playing
- Based on various weather patterns
- The decision tree algorithm generates a Tree of questions that leads to a golf decision.



Attributes				Classes
Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

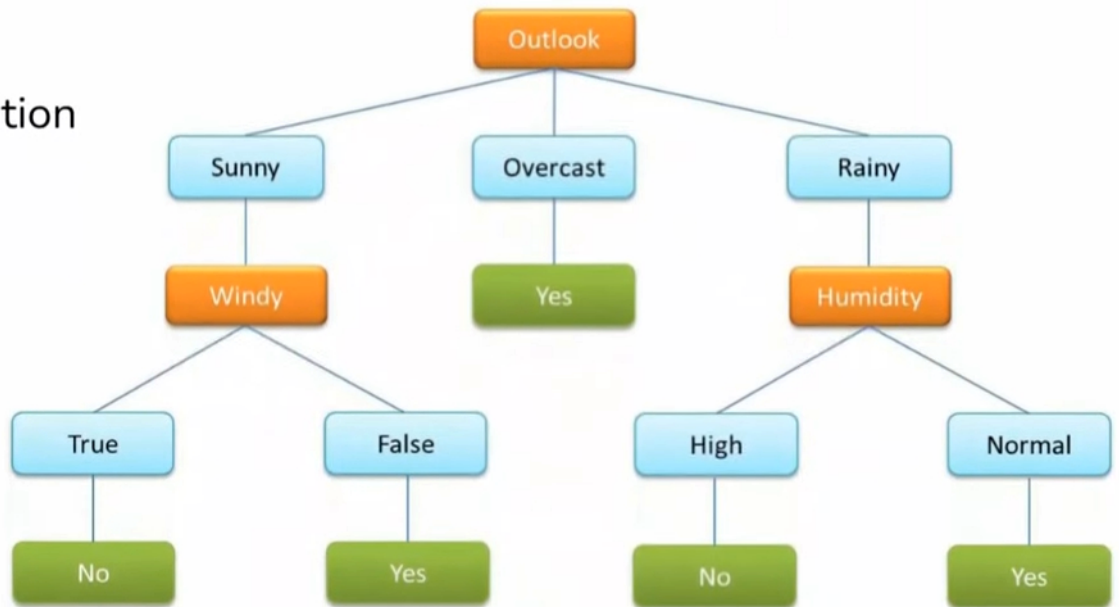
At each layer of the tree

- The decision tree picks an attribute/feature to move onto the next layer.
- The natural question arises
- How do we know which feature should be picked first?
- Answer:

The feature that gives us the most information

We already know how to calculate the information gained from each feature.

At each layer, we simply pick the feature with the highest information Gain.



Which features should we pick first?

Given the data table, we can calculate the conditional entropy for each feature. The original entropy, as we previously calculated, is

$$H(Y) = 0.94. \quad (1)$$

The conditional entropy are

$$H(Y|\text{outlook}) = 0.693 \quad (2)$$

$$H(Y|\text{Temp}) = 0.911 \quad (3)$$

$$H(Y|\text{Humidity}) = 0.788 \quad (4)$$

$$H(Y|\text{Windy}) = 0.892. \quad (5)$$

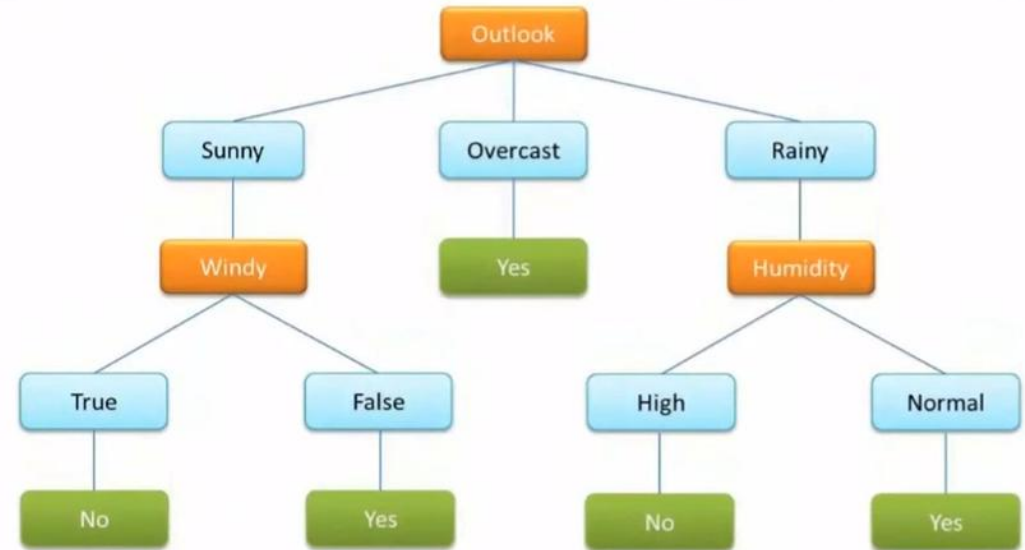
The information gained then becomes

$$\Delta H_{Y|\text{outlook}} = 0.94 - 0.693 = 0.247 \quad (6)$$

$$\Delta H_{Y|\text{Temp}} = 0.94 - 0.911 = 0.029 \quad (7)$$

$$\Delta H_{Y|\text{Humidity}} = 0.94 - 0.788 = 0.152 \quad (8)$$

$$\Delta H_{Y|\text{Windy}} = 0.94 - 0.892 = 0.048. \quad (9)$$



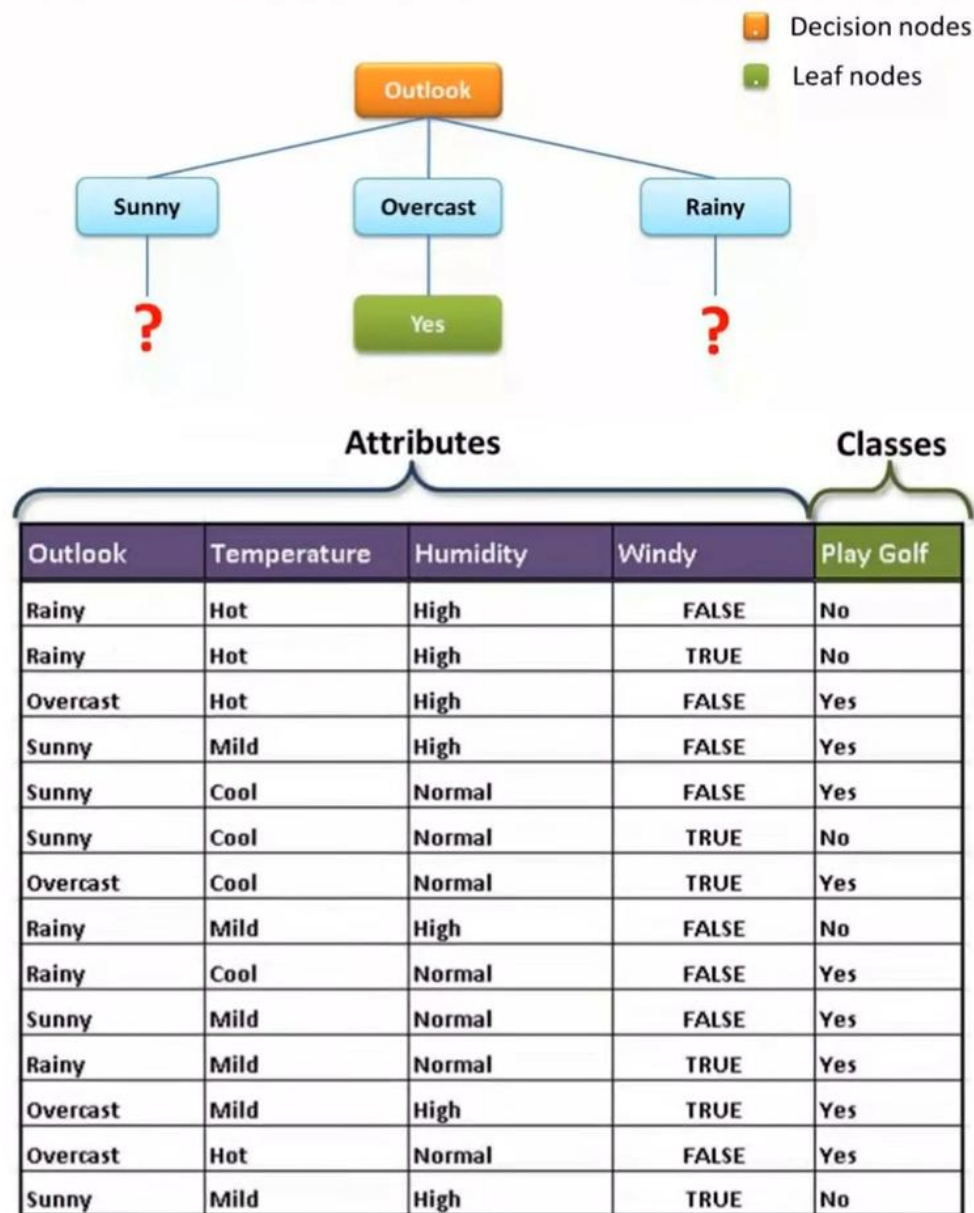
After splitting the branch.

Notice that after picking Outlook as the 1st feature to split the tree.

- Having the outlook=overcast, only has 1 possible golf outcome, Yes.
- As the tree splits, if there is only 1 possible outcome, you can safely end the branch of the tree with the label.
- As the tree splits, you have to split the branches further with other features if there are multiple possible outcomes.
- For each new branch. You will assume that the outlook is already known and shrink the table to calculate the next branch.
- For example, let's assume we are working on the Rainy branch. Then to identify the next feature to use, we will use a smaller table where the outlook is only Rainy.

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No

Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes



Practice, Practice

After identifying the **outlook** as the first question. We need to identify the next branch. Let's **Playing Golf** be Y and the other features be X . (use \log_e)

- Look at the Rain Status table.
- Identify the next feature that provides the largest information gain.



Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No

Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Solution to the next Layer

Golf	Yes	No	Temp/Golf	Yes	No	Prob	Humidity/Golf	Yes	No	windy/Golf	Yes	No
	2	3	Hot	0	2	2/5	High	0	3	True	1	1
			Mild	1	1	2/5	Normal	2	0	False	1	2
			Cool	1	0	1/5						

$$H(Y) = -2/5 \log(2/5) - 3/5 \log(3/5) = 0.67$$

$$\begin{aligned}
 H(Y|T) = & -p(t = \text{hot}) [p(y = \text{yes}|t = \text{hot}) \log_2(p(y = \text{yes}|t = \text{hot})) + p(y = \text{no}|t = \text{hot}) \log_2(p(y = \text{no}|t = \text{hot}))] \\
 & - p(x = \text{cold}) [p(y = \text{yes}|t = \text{cold}) \log_2(p(y = \text{yes}|t = \text{cold})) + p(y = \text{no}|t = \text{cold}) \log_2(p(y = \text{no}|t = \text{cold}))] \\
 & - p(x = \text{mild}) [p(y = \text{yes}|t = \text{mild}) \log_2(p(y = \text{yes}|t = \text{mild})) + p(y = \text{no}|t = \text{mild}) \log_2(p(y = \text{no}|t = \text{mild}))]
 \end{aligned}$$

$$H(Y|T) = -\frac{2}{5} [0 + 0] - \frac{2}{5} \left[\frac{1}{2} \log\left(\frac{1}{2}\right) + \frac{1}{2} \log\left(\frac{1}{2}\right) \right] - \frac{1}{5} [0 + 0] = 0.277$$

$$H(Y|H) = 3/5(0) + 2/5(0) = 0$$

$$H(Y|W) = -2/5 \left[\frac{1}{2} \log\left(\frac{1}{2}\right) + \frac{1}{2} \log\left(\frac{1}{2}\right) \right] - 3/5 \left[\frac{1}{3} \log\left(\frac{1}{3}\right) + \frac{2}{3} \log\left(\frac{2}{3}\right) \right] = 0.659$$

This implies that Humidity gives us the most information Gain

$$\Delta H_{Y|T} = 0.67 - 0.277 = 0.393$$

$$\Delta H_{Y|H} = 0.67 - 0 = \mathbf{0.67}$$

$$\Delta H_{Y|W} = 0.67 - 0.659 = 0.011$$

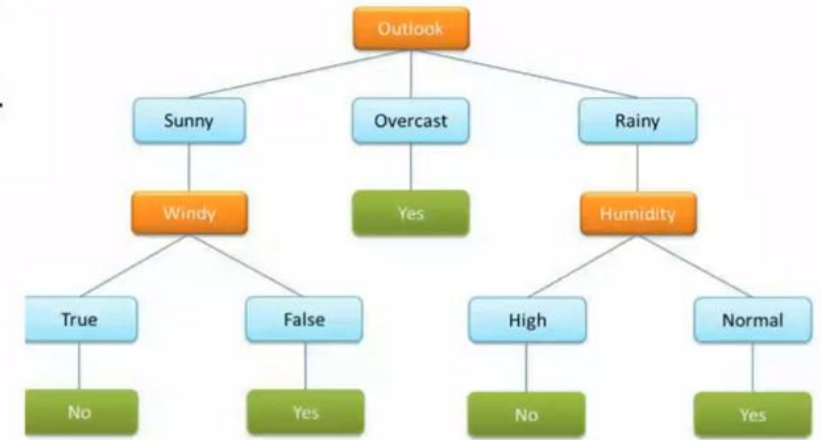
Decision Tree is a very popular algorithm

For a long time, decision tree has been one of the most popular classification algorithms.

- It is easy to implement in Python.
- It generally does a good job on simpler datasets.
- It tells you which features are most important.
- In fact, it tells you the importance of the features in order of importance based on the information gained.
- It is commonly used with Ensemble methods like bagging to future improve its performance.

Next up

- what are Ensemble methods?
- what is bagging?
- how does bagging future a tree into a forest?



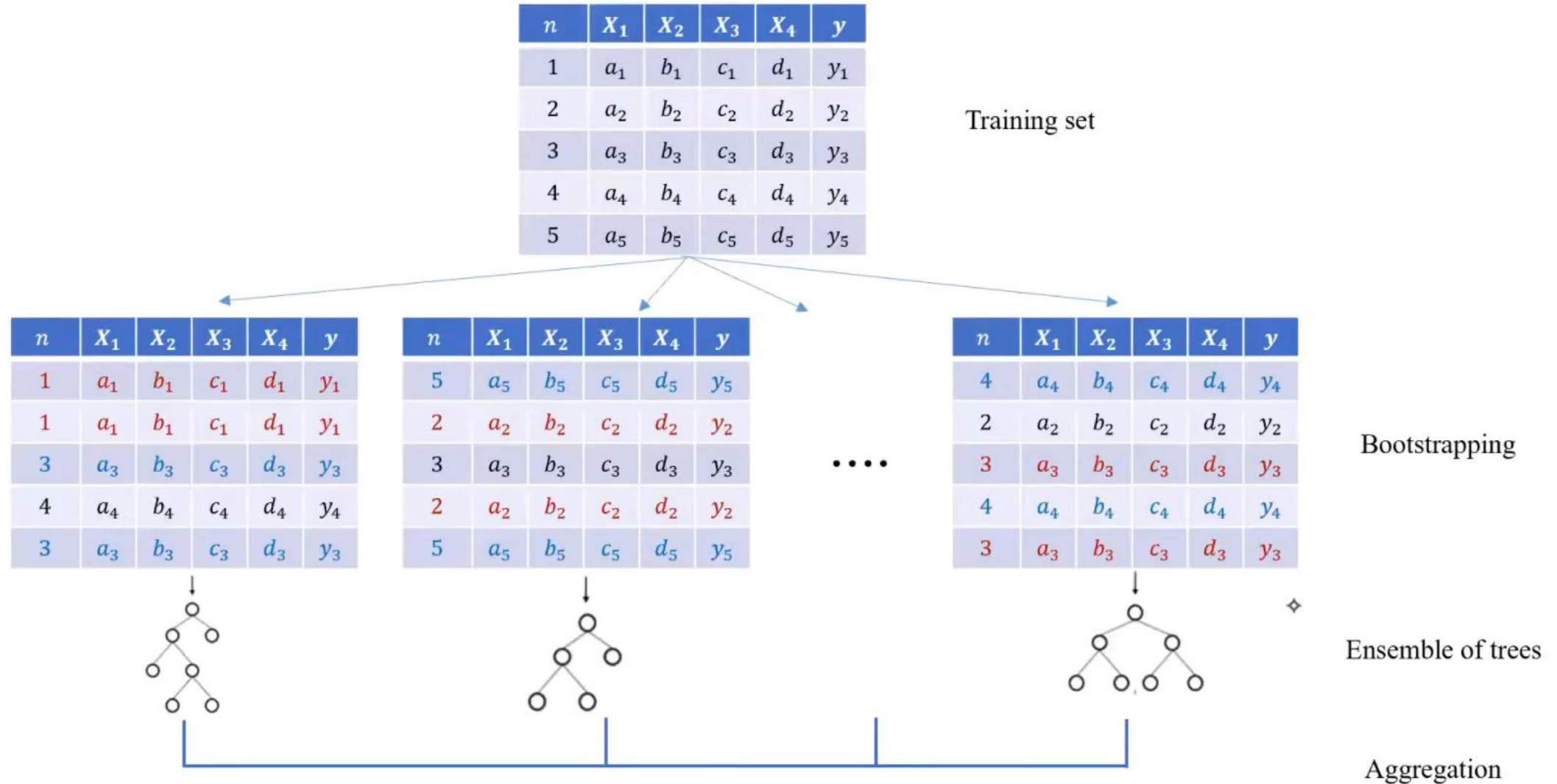
Bagging and Random Forest

- In bagging, you use a random subset of the data and train the classifier.
- Once you train a classifier, you again use a different random subset of the data to train the same classifier.
- You repeat this process n times to obtain n different classifiers.
- Because different data is used each time, each classifier might give a different result.
- Given different classification results from n classifiers, we use **majority vote** to determine the final result.

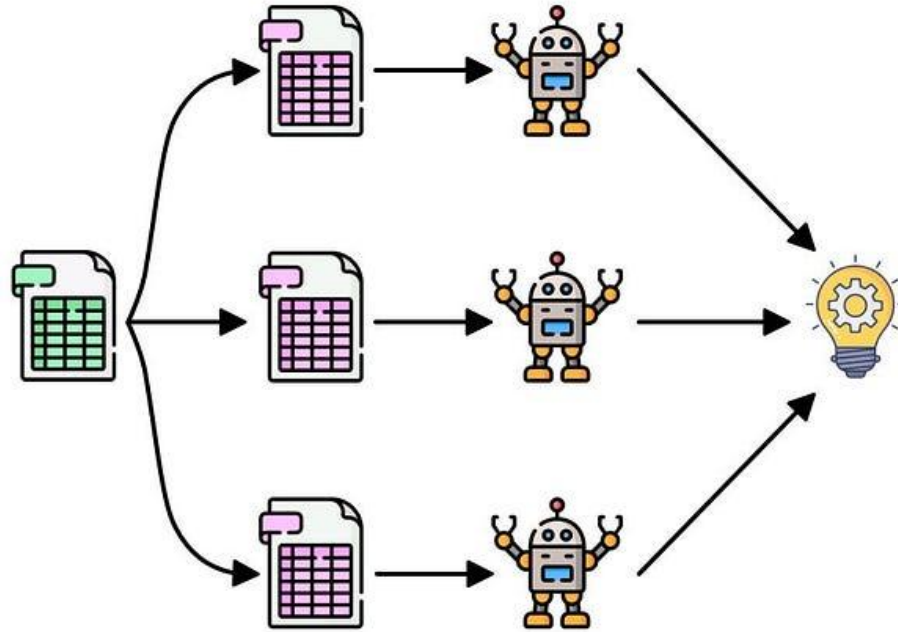
Using bagging on decision tree is called **random forest**.

- since you are training multiple trees
- the trees become a forest.
- Since you are randomly picking the data to train a tree
- That is why we call it a random forest.
- In general, a random forest performs better than a single tree.

Bootstrap Aggregation (Bagging)

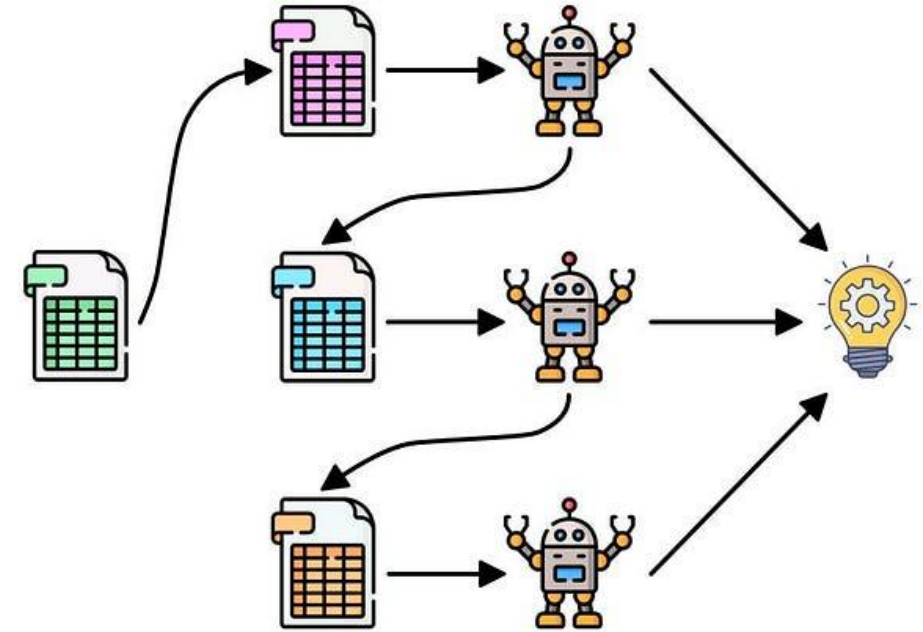


Bagging



Parallel

Boosting

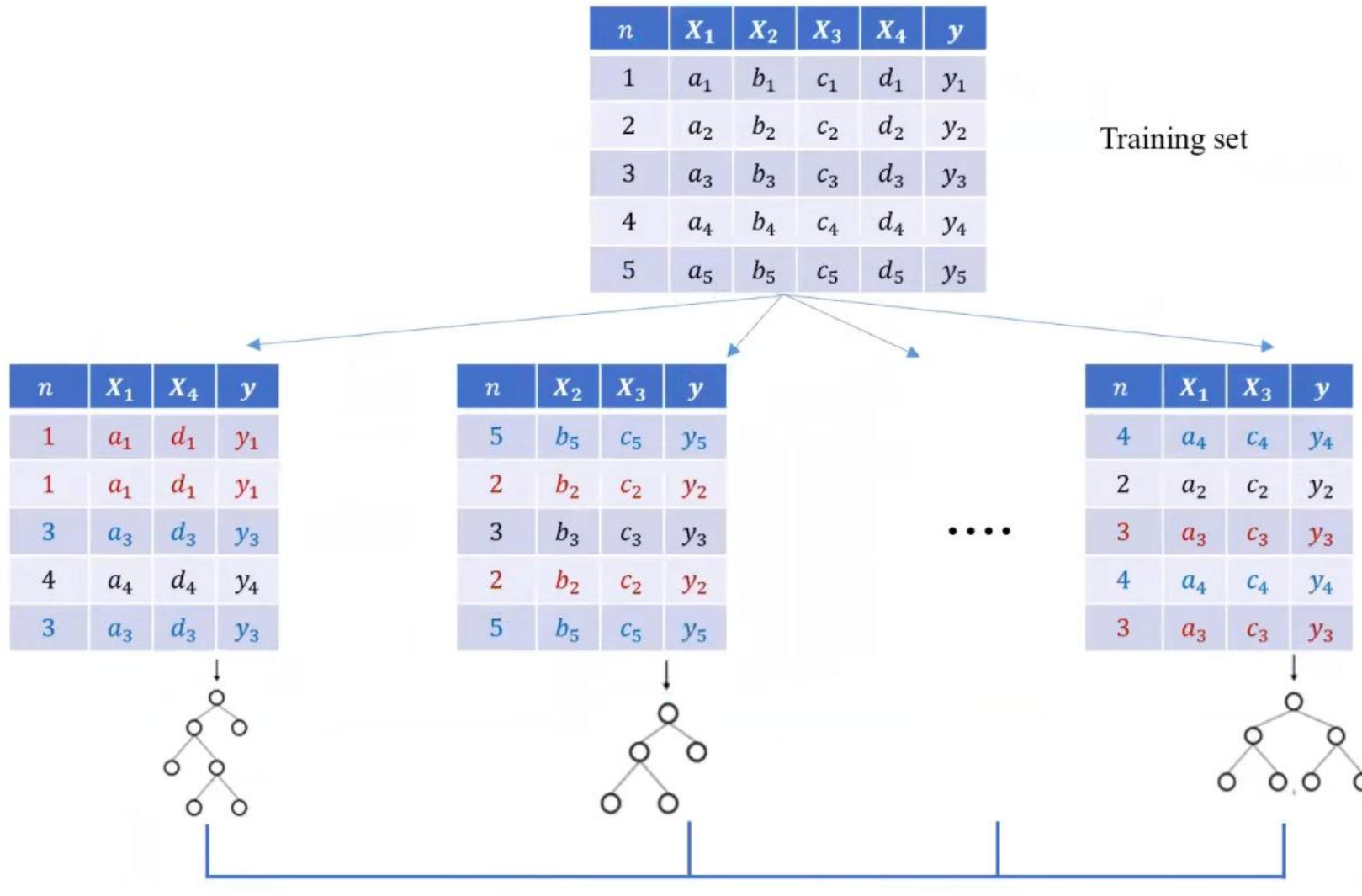


Sequential

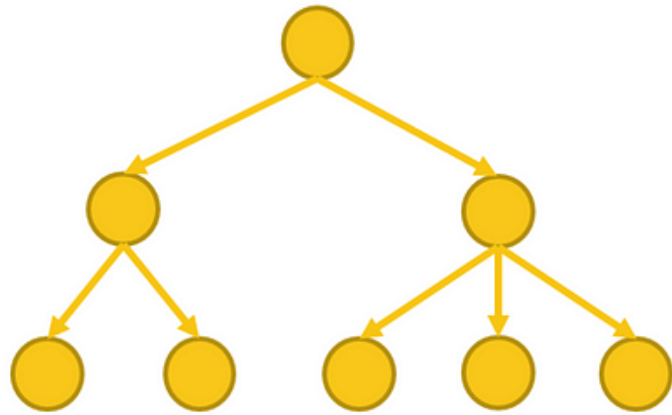
Random Forests



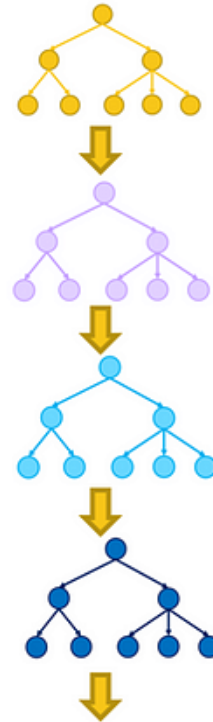
Random subset of features happens at each split!



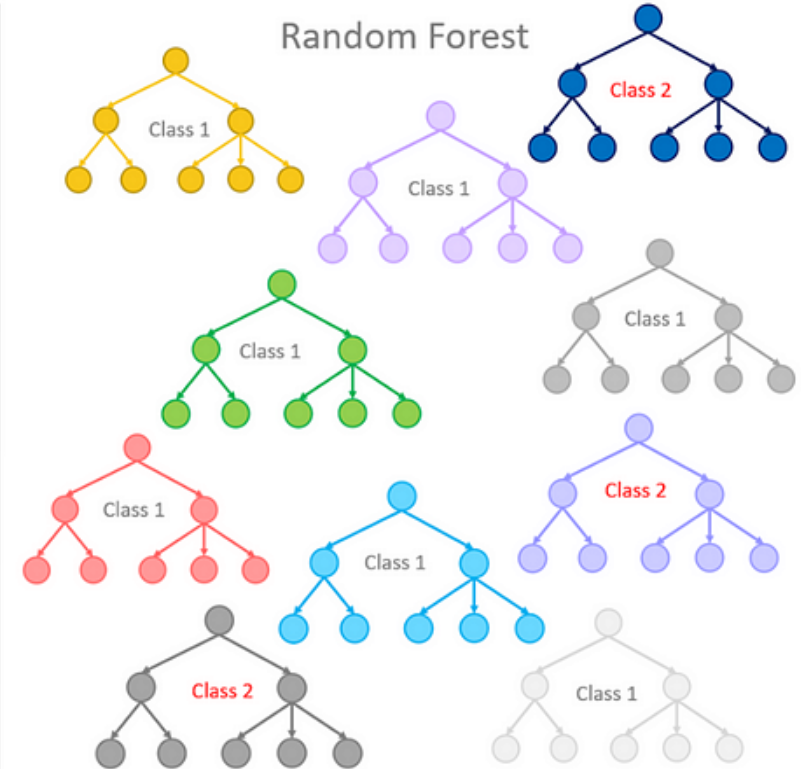
Single Decision Tree



Gradient Boosted Trees

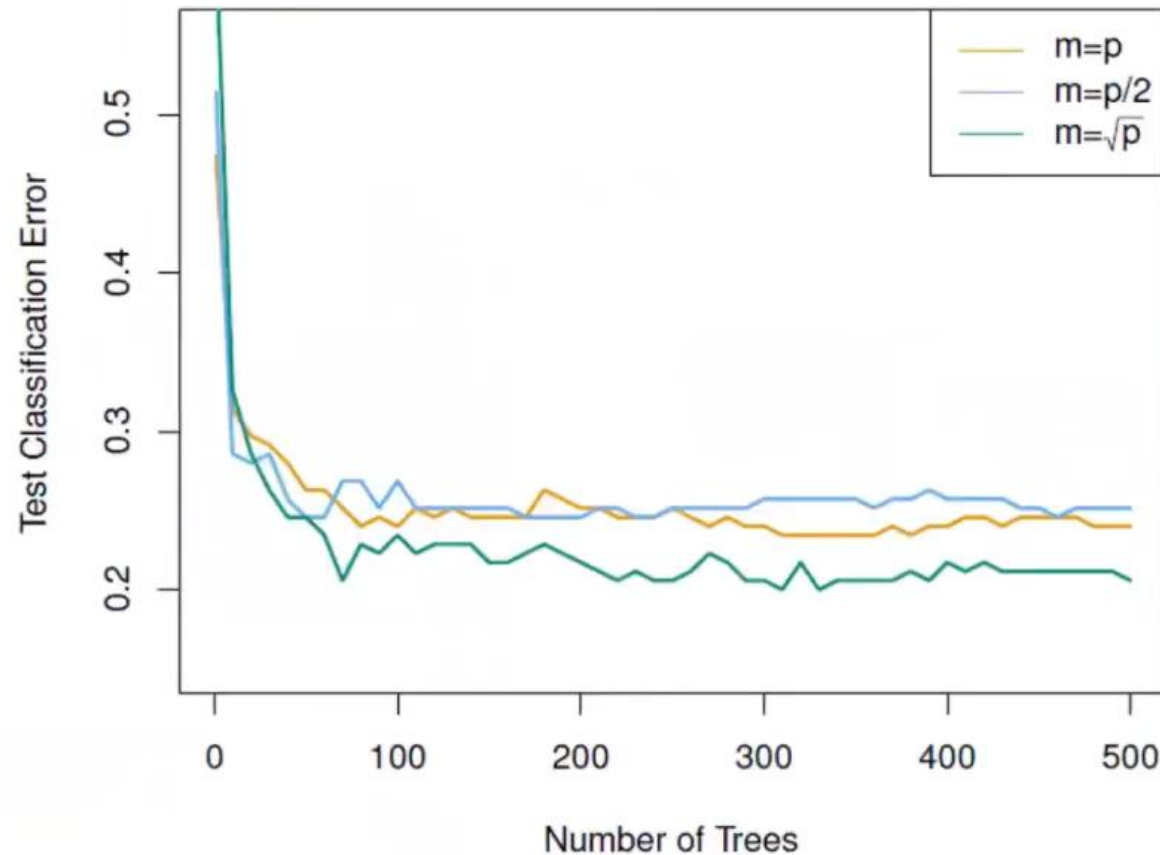


Random Forest



Random Forests vs Bagging

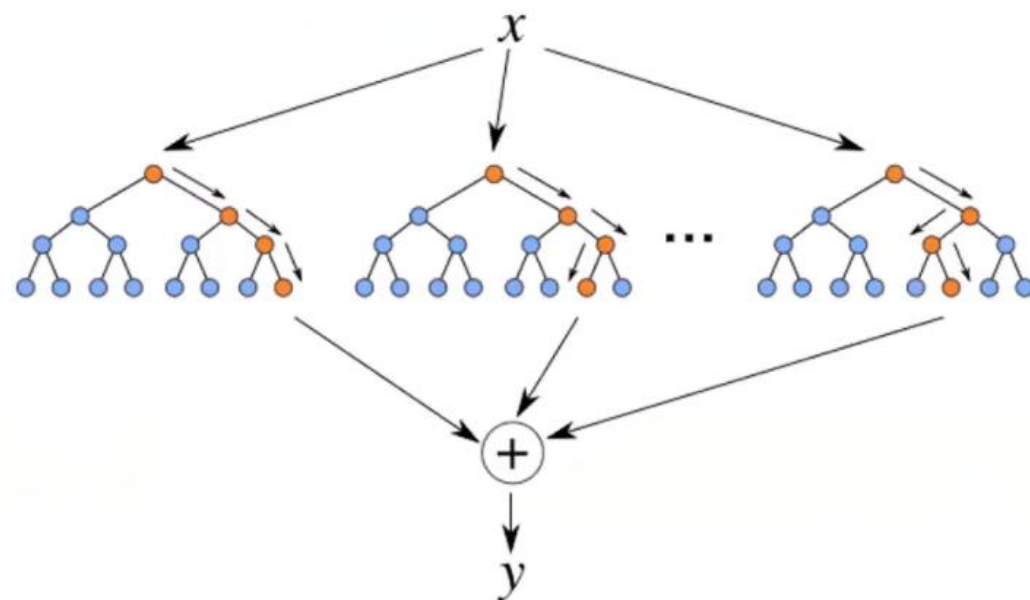
- A random subset \mathbf{m} , can be any subset of \mathbf{p} features like $\frac{p}{2}$ or \sqrt{p} or $\log(p)$ etc.



Hyperparameters

- The most important hyper parameters are:

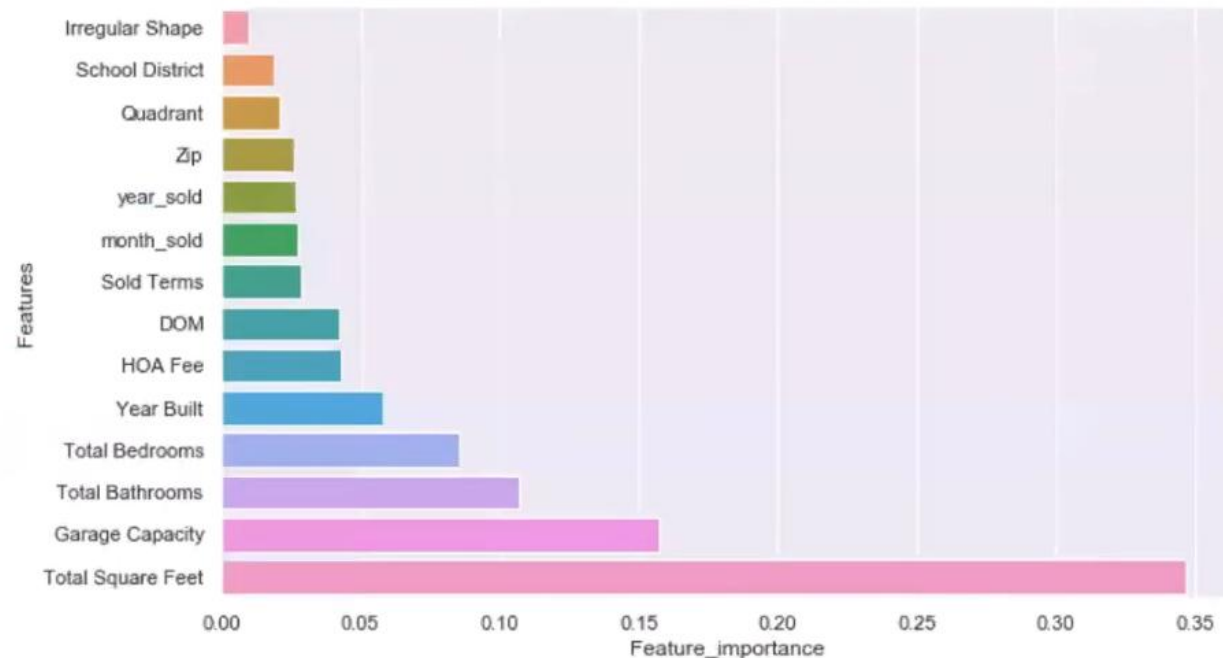
- ✓ The number of subset features (**m**)
- ✓ The number of trees to use (**B**)
- ✓ The minimum **size** of each node (or leaf)
- ✓ The maximum **number** of leaf nodes
- ✓ The maximum **depth** of each tree
- ✓ Criterion: gini, entropy



- Grid search CV could be used to tune a combination of hyper parameters.

Feature importance

- **Feature importance** refers to techniques that assign a **score** to input features based on how **useful** they are at predicting a target variable.
- For RF , the total amount that the **RSS is decreased** / **Gini index or entropy is decreased** due to splits over a given predictor, averaged over all B trees. A **large** value indicates an **important** predictor.

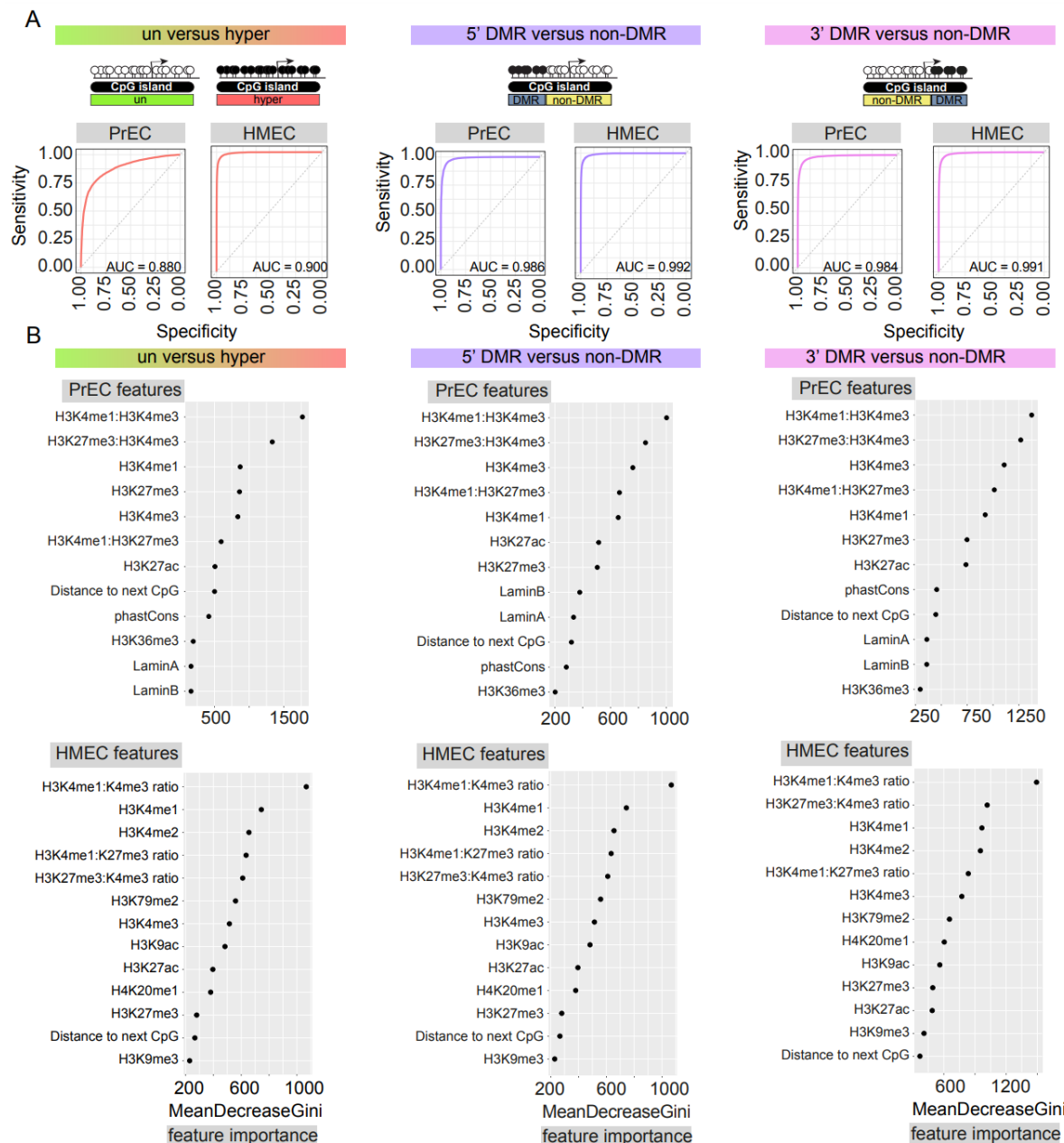


ARTICLE · Volume 35, Issue 2, P297-314.E8, February 11, 2019 · [Open Archive](#)

[Download Full Issue](#)

DNA Hypermethylation Encroachment at CpG Island Borders in Cancer Is Predisposed by H3K4 Monomethylation Patterns

Ksenia Skvortsova^{1,15} · Etienne Masle-Farquhar² · Phuc-Loi Luu¹ · Jenny Z. Song¹ · Wenjia Qu¹ · Elena Zotenko¹ · Cathryn M. Gould¹ · Qian Du¹ · Timothy J. Peters¹ · Yolanda Colino-Sanguino¹ · Ruth Pidsley¹ · Shalima S. Nair¹ · Amanda Khoury¹ · Grady C. Smith¹ · Lisa A. Miosge³ · Joanne H. Reed² · James G. Kench^{4,5,14} · Mark A. Rubin^{6,7,8,9,10} · Lisa Horvath^{5,11,12,13,14} · Ozren Bogdanovic^{11,15} · Sue Mei Lim^{16,17,18} · Jose M. Polo^{16,17,18} · Christopher C. Goodnow^{2,11} · Clare Stirzaker^{1,11,19}  · Susan J. Clark^{1,11,19,20}  [Show less](#)



Hands-on Exercises

- Decision Tree
 - <https://www.geeksforgeeks.org/decision-tree-introduction-example/>
 - Apply the decision on the prostate cancer methylation data
- Random Forest and Feature Importance
 - https://inria.github.io/scikit-learn-mooc/python_scripts/dev_features_importance.html
 - <https://www.kaggle.com/code/pratikkghandhi/predicting-breast-cancer-with-random-forest-95>

Thanks for your attention!

Giảng viên: TS. Lưu Phúc Lợi

Luu.p.loi@gmail.com

Zalo: 0901802182