Random Forest
And Variable Importance

CS1090A Introduction to Data Science

Pavlos Protopapas, Natesh Pillai, Chris Gumb

Dylan Wu
Arashiyama Forest, Kyoto

Outline

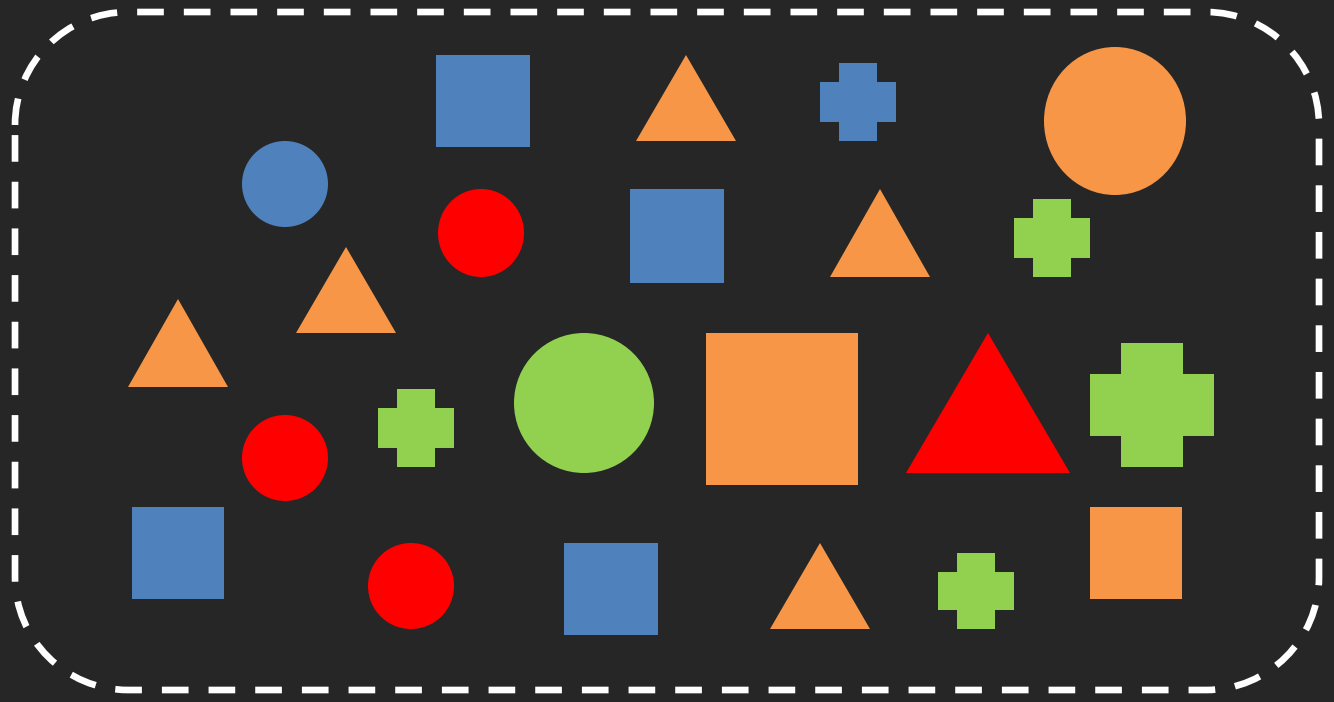
- Motivation
- Random Forest
- Variable Importance
- Missing Data (again)
- Class Imbalance
- Tree building algorithms

Outline

- Motivation
- Random Forest
- Variable Importance
- Missing Data (again)
- Class Imbalance
- Tree building algorithms

Motivation

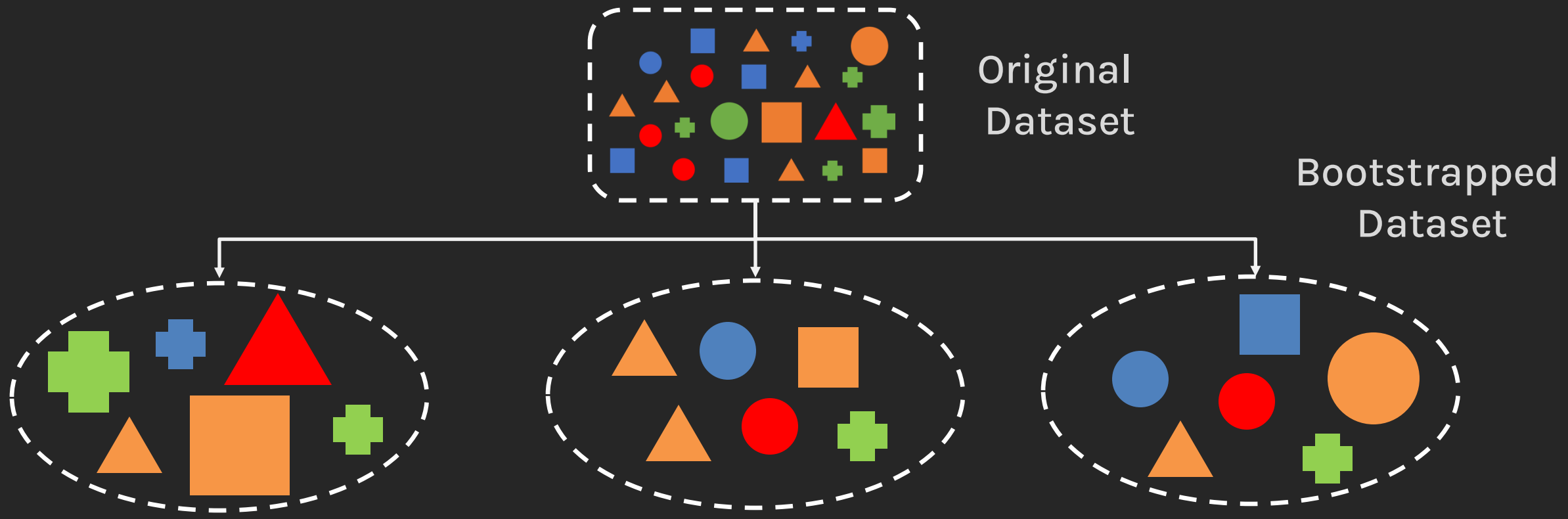
Consider a dataset of geometric shapes with features of **Color**, **Shape** and **Size**, and we want to use the ensemble learning method, Bagging (Bootstrap Aggregating).



Original Dataset

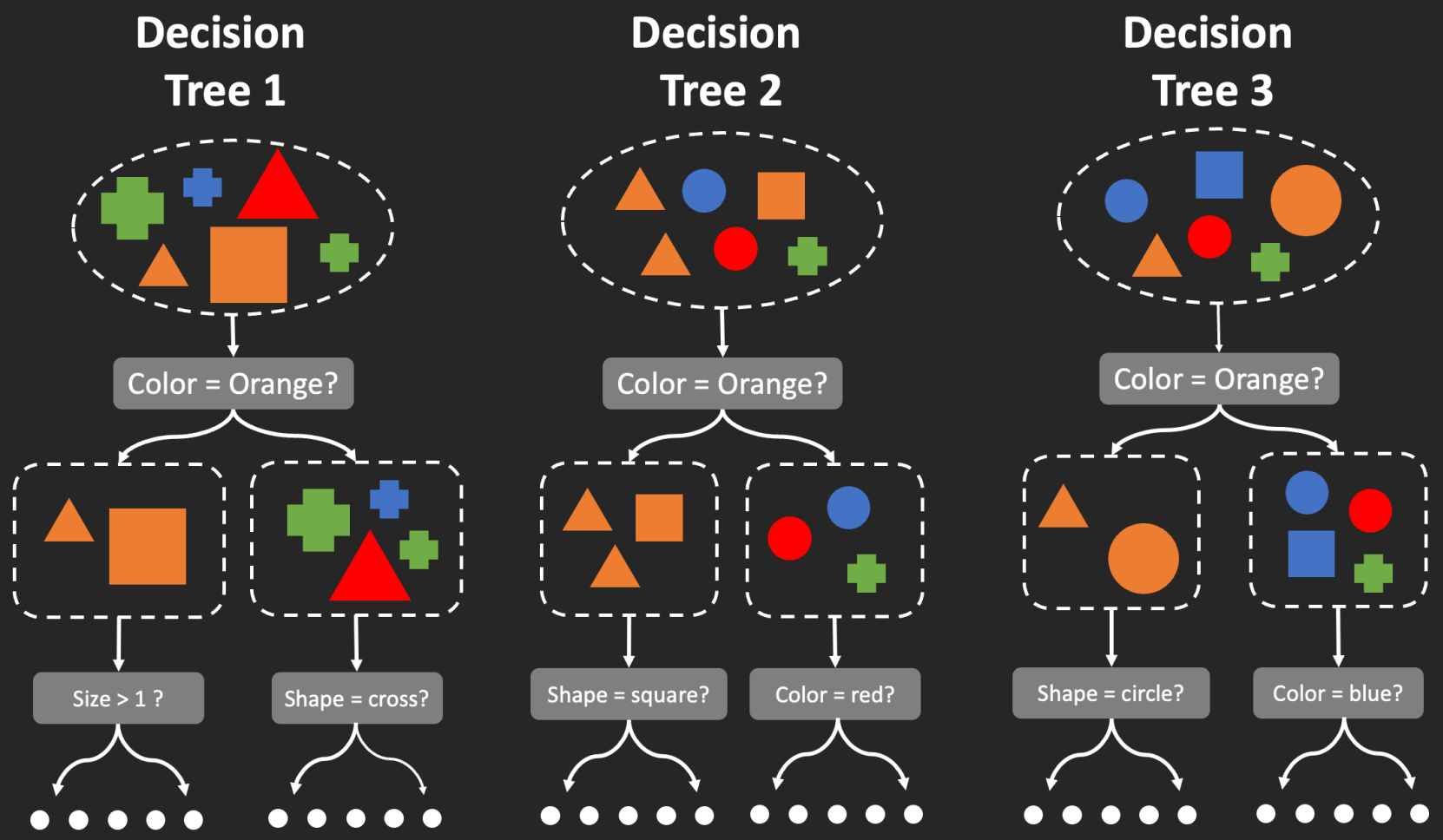
Motivation

Multiple subsets of the dataset are created using bootstrapping in the first step of Bagging.



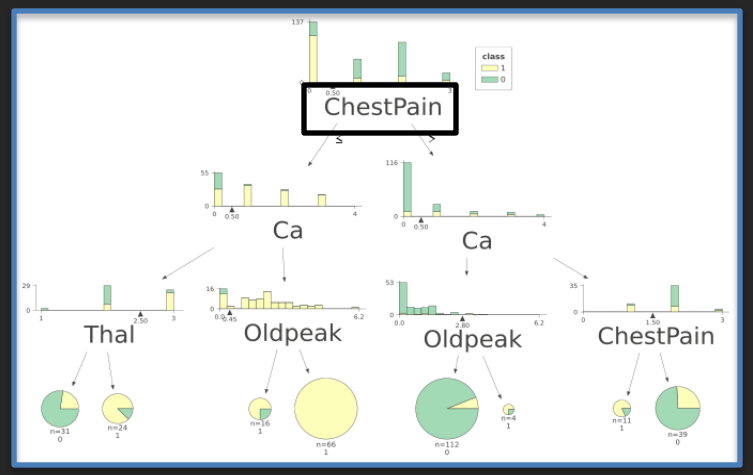
Motivation

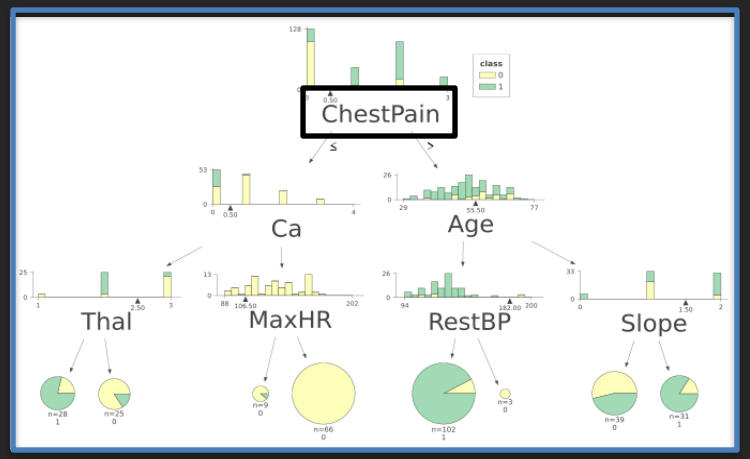
Then multiple decision trees are fitted with each of the bootstrapped datasets.

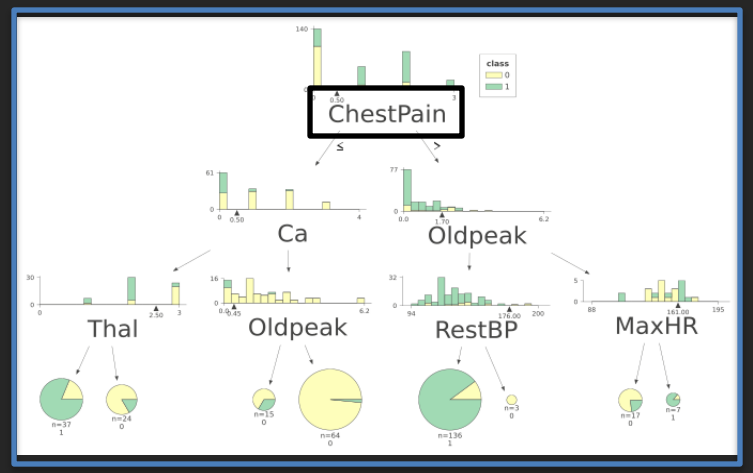


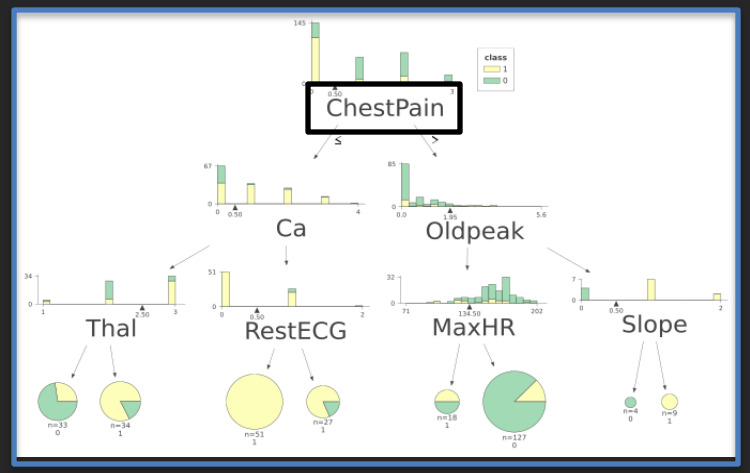
Motivation

But as we have seen, these trees are correlated.



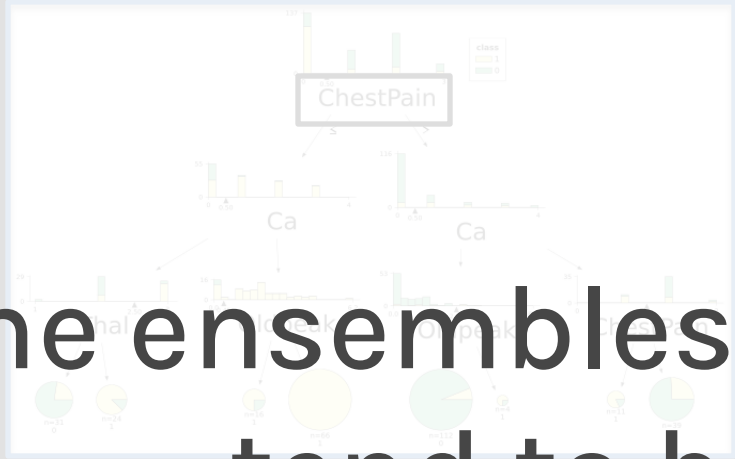


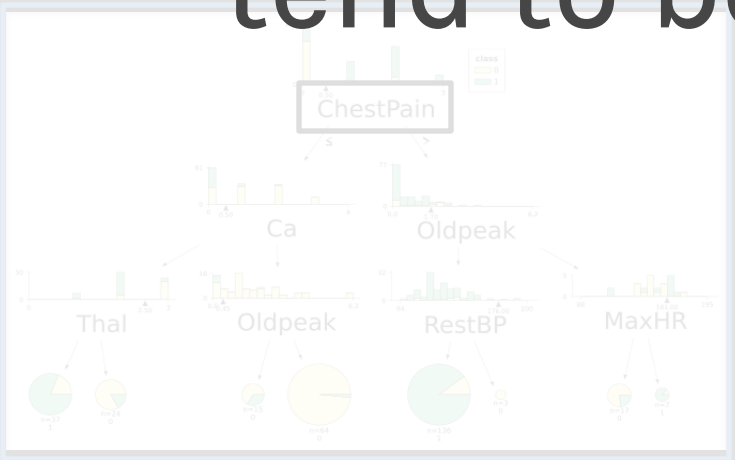


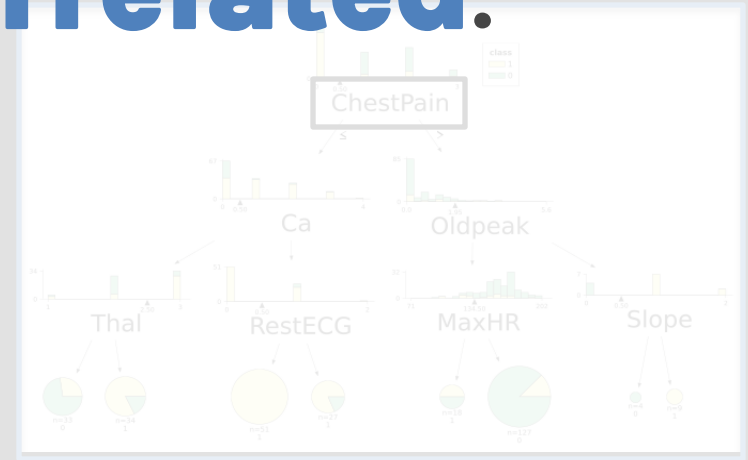


Motivation

Consider the following decision trees in a bagging model that predicts if a person has heart disease:



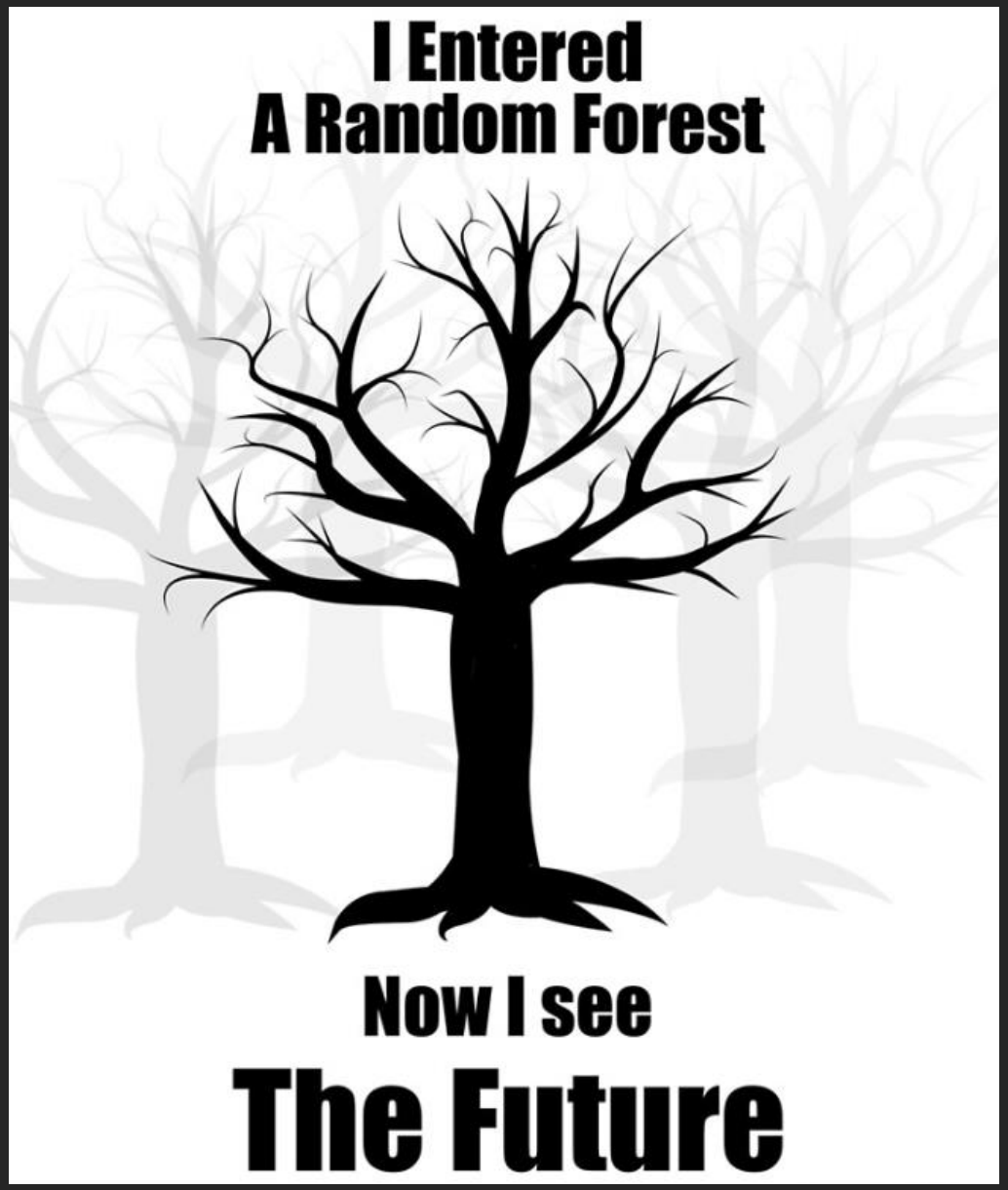




The ensembles of trees in bagging tend to be **correlated**.

Outline

- Motivation
- **Random Forest**
- Variable Importance
- Missing Data (again)
- Class Imbalance
- Tree building algorithms


I Entered
A Random Forest
Now I see
The Future

Random Forest

How can we avoid this issue then?

OR How do we de-correlate the trees?

Random forest is a modified form of bagging that creates ensembles of **independent** decision trees.

Then the question is, how?

Quiz time

Which of the following(s) approaches do you think could be effective in reducing correlation among the trees?

Options

A. Limit the depth of each tree in a unique way.

B. Ensure that each tree's split does not include any of the splits used in the previous trees.

C. Randomly select a subset of predictors to consider for splitting at each node.

D. Randomly choose a single predictor for every split.

E. Design the splits such that the probability of choosing a significant predictor is low, leading to an ensemble of weaker models.

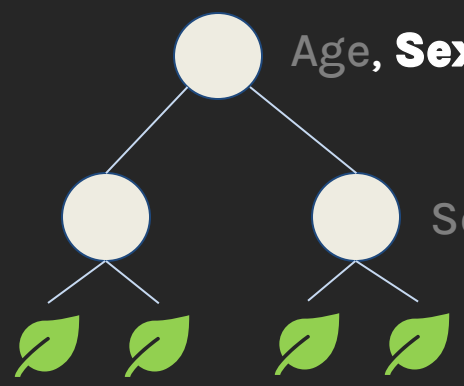
Random Forest

Consider a dataset that contains the following predictors:

Age, Sex, Max HR,
Cholesterol, Chest Pain

Bootstrap, B_1

Randomly choose **a subset of predictors** and find **the best predictor** to split on and its threshold

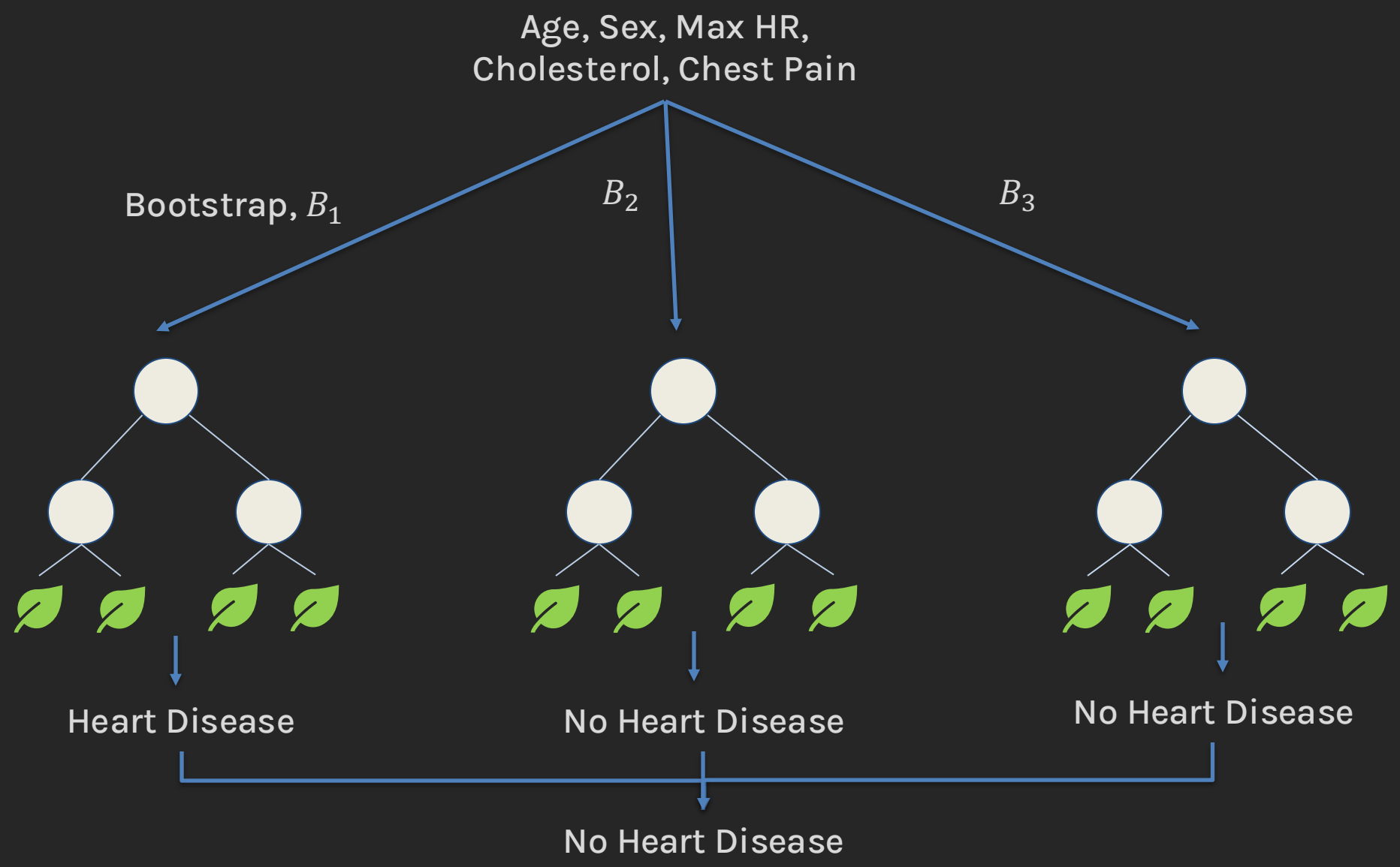


Age, **Sex**, Cholesterol

Cholesterol,
Max HR,
Chest Pain

Sex, Max HR, **Chest Pain**

Random Forests



Random Forests

In summary, Random Forest works in the following ways:

1. Create B **bootstrapped datasets** with all J predictors (same as in bagging).

2. Initialize a random forest with B decision trees.

3. For each tree, at each split, we **randomly** select a **subset** of J' predictors from the full set of predictors ($J' < J$).

4. Amongst the J' **predictors**, we select the optimal predictor and the optimal threshold for the corresponding split.

Tuning Random Forests

Random forest models have multiple **hyper-parameters** to tune:

1. The **number of predictors** to randomly select at each split.
2. The **total number of trees** in the ensemble.
3. The **stopping criteria** - maximum depth, minimum leaf node size, etc.
4. The **splitting criterium** – gini, entropy

Tuning Random Forests

There are standard (default) values for each of random forest hyper-parameters recommended by long time practitioners.

For **THE NUMBER OF PREDICTORS**

$\sqrt{N_j}$ predictors for classification

$\frac{N_j}{3}$ predictors for regression

For **THE NUMBER OF TREES**, we use out-of-bag errors. With OOB, training and validation can be done in a single sequence - once the out-of-bag error stabilizes, adding more trees may not be beneficial.

Also, generally these parameters should be tuned through **OOB** (making them data and problem dependent).

Outline

- Motivation
- Random Forest
- **Variable Importance**
- Missing Data (again)
- Class Imbalance
- Tree building algorithms

Variable Importance for RF (and Bagging)

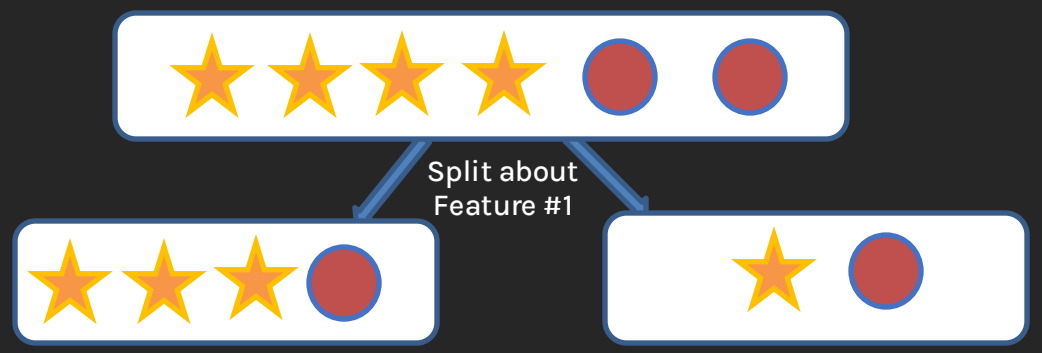
1. Mean Decrease in Impurity.
2. Permutation importance.
3. LIME, SHAP values [lab and reading]

Variable Importance for RF (and Bagging)

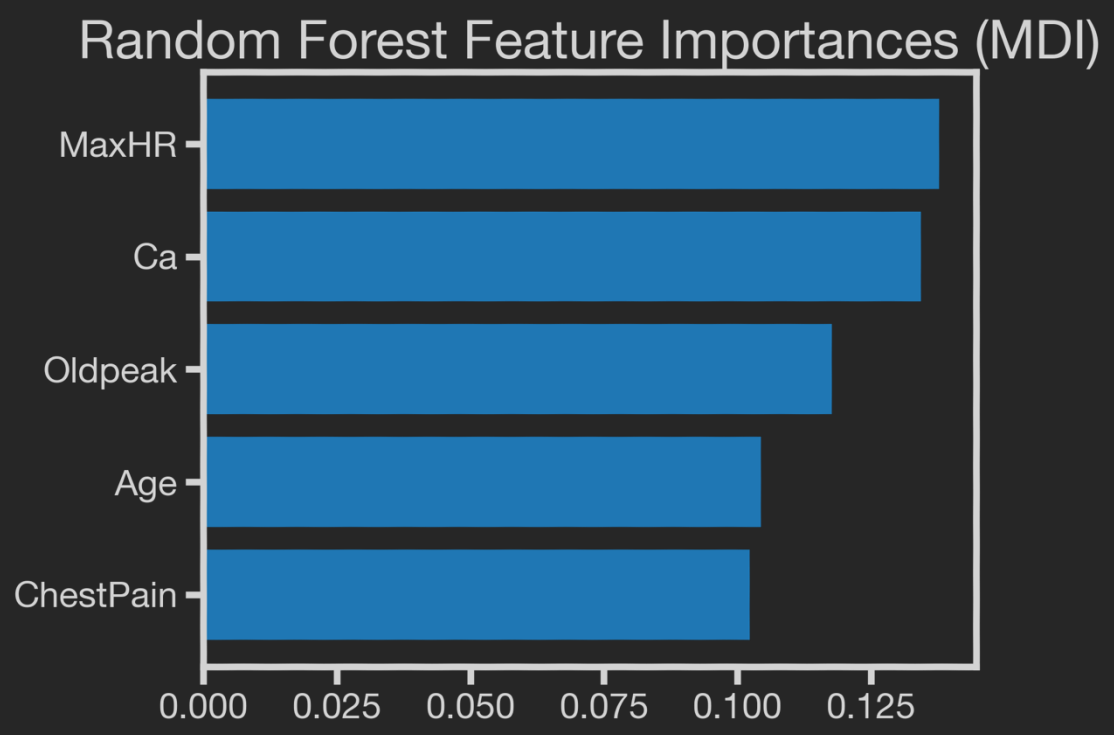
1. **Mean Decrease in Impurity.**
2. Permutation importance.
3. LIME, SHAP values [lab and reading]

Mean Decrease in Impurity (MDI)

Decision trees make splits that maximize the decrease in impurity. By calculating the mean decrease in impurity for each feature across all trees, we arrive at the variable importance for a bagging or random forest model.

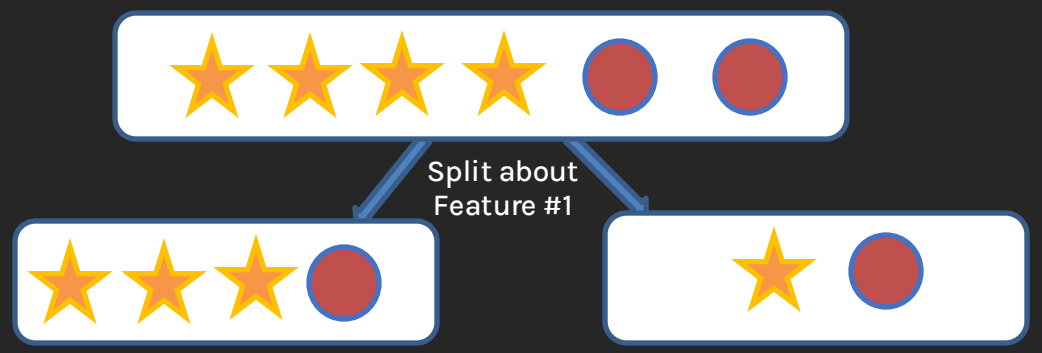


Average this over
all the trees

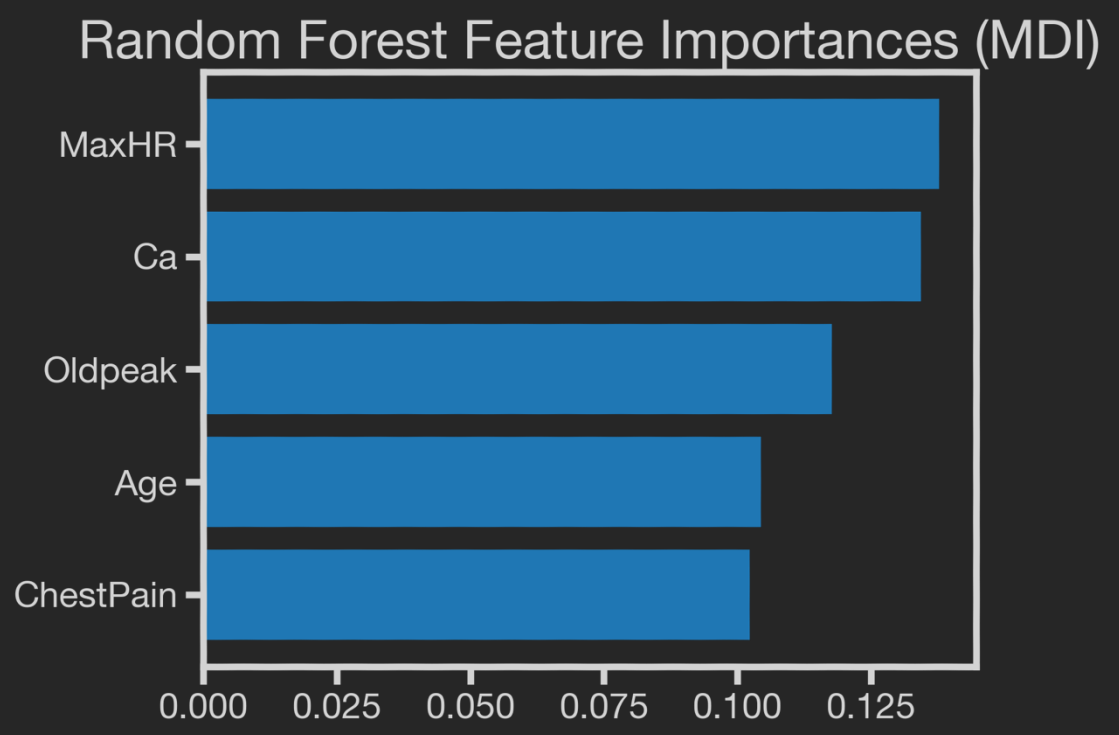


Mean Decrease in Impurity (MDI)

Instead of calculating at the individual tree level, we can calculate the MEAN decrease in impurity for each predictor across all trees.

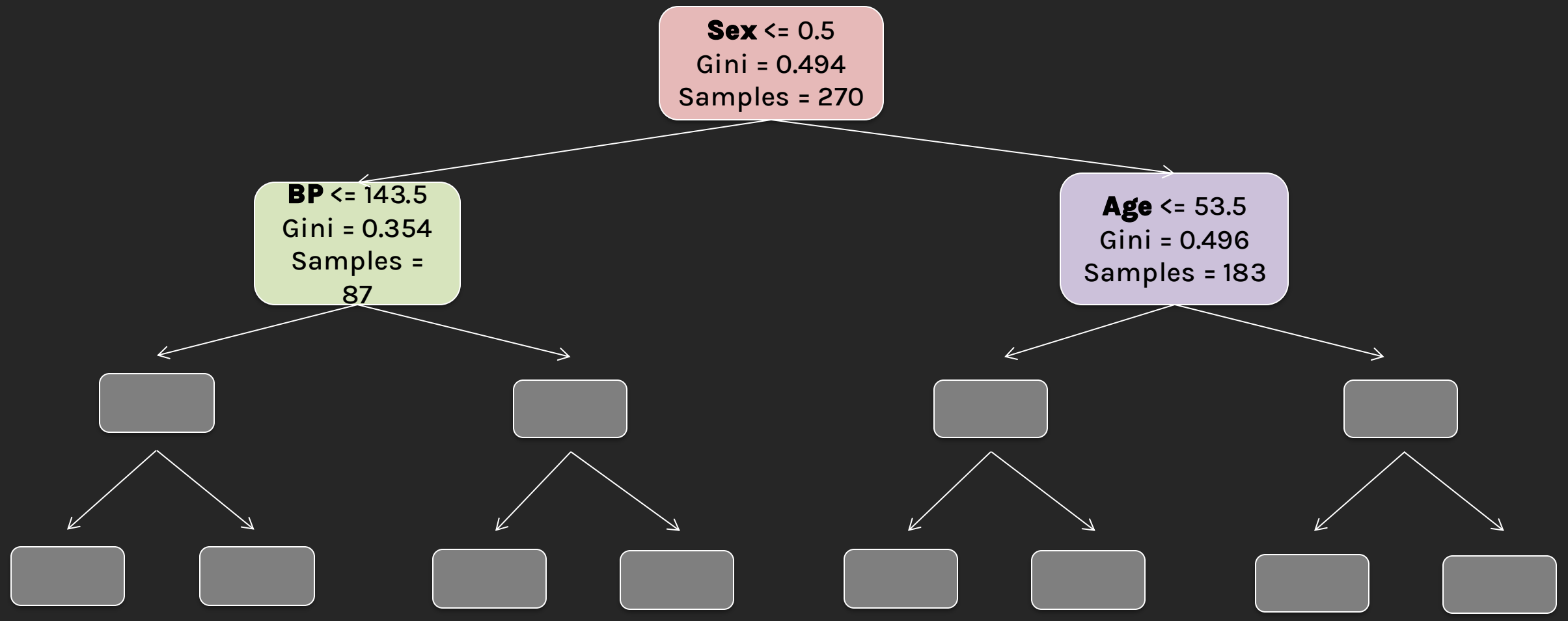


Average this over
all the trees



Mean Decrease in Impurity (MDI)

Consider the following decision tree:



Mean Decrease in Impurity (MDI)

Step 1: Calculate the mean decrease in impurity for each node q in the decision tree.

Mean decrease in impurity for each node q

$$\Delta I_q = \left(\frac{n}{N}\right)\left[Gini_n - \sum_{m \in \text{Child(n)}} \left(\frac{m}{n}\right) Gini_m\right]$$

Fraction of n samples from the node out of the **whole dataset**

Sum over all children of node n

Mean Decrease in Impurity (MDI)

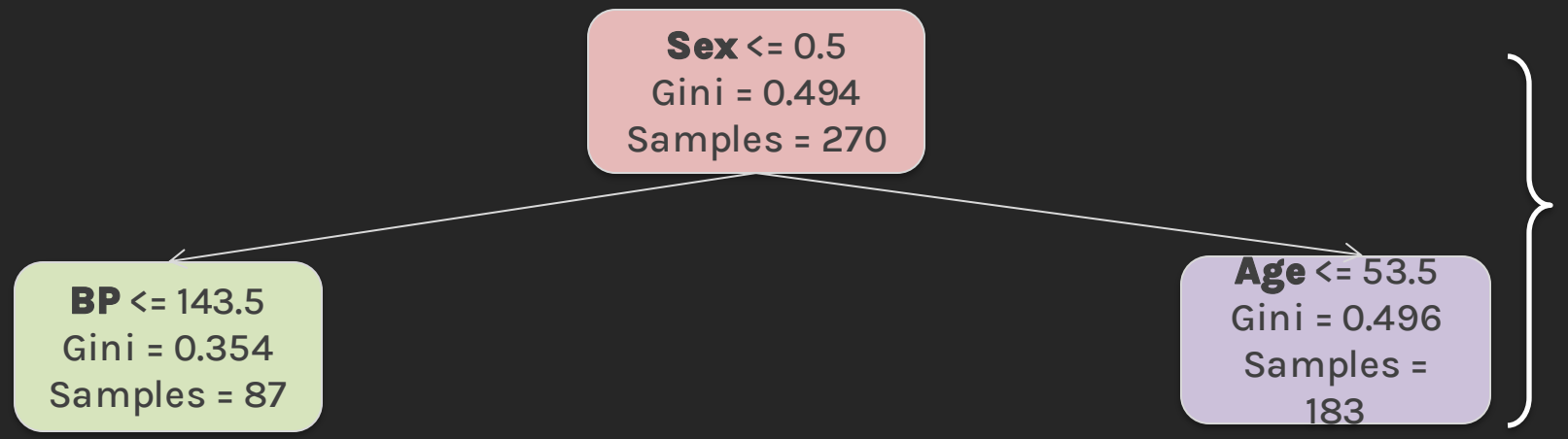
Step 1: Calculate the mean decrease in impurity for each node q in the decision tree.

$$\Delta I_q = \left(\frac{n}{N}\right)\left[Gini_n - \left(\frac{m_L}{n}\right) Gini_{m_L} - \left(\frac{m_R}{n}\right) Gini_{m_R}\right]$$

Mean Decrease in Impurity (MDI)

$$\Delta I_q = \left(\frac{n}{N}\right)\left[Gini_n - \sum\left(\frac{m}{n}\right) Gini_m\right]$$

Step 1: Calculate the mean decrease in impurity for each node q in the decision tree.



Impurity Decrease

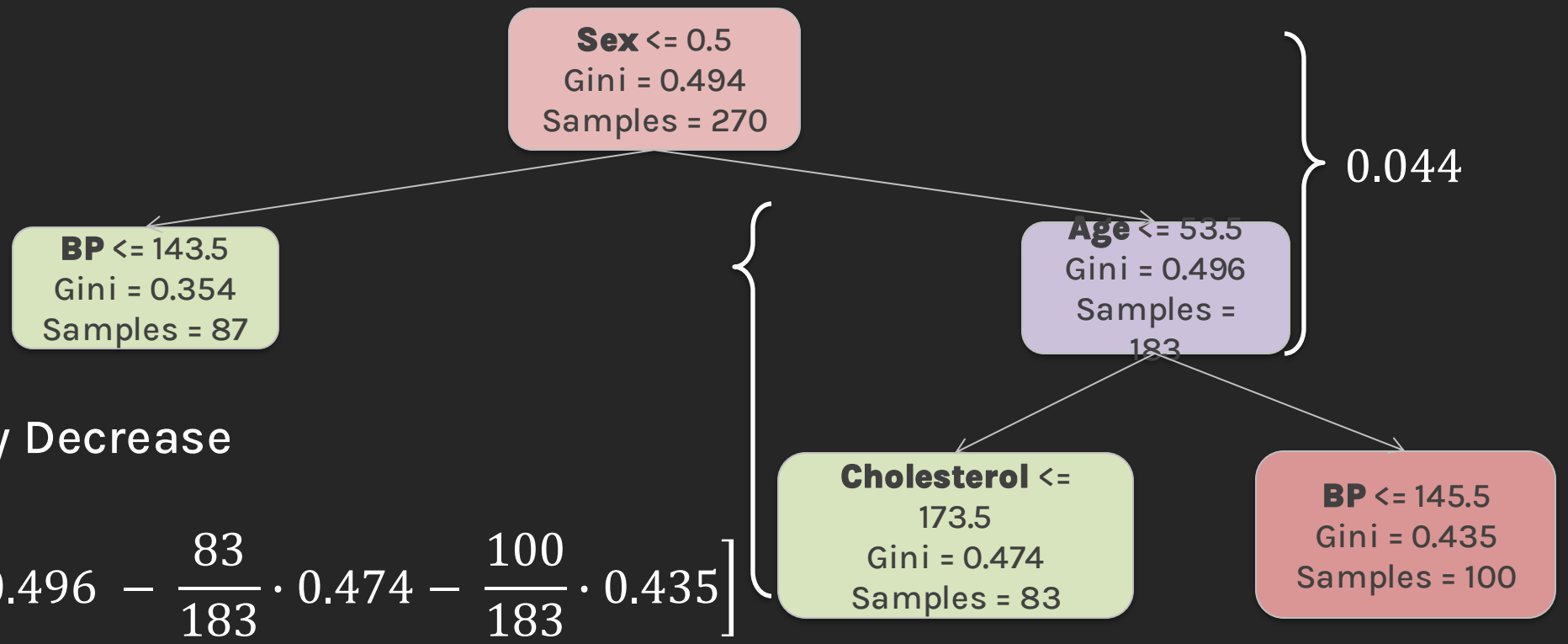
$$= \frac{270}{270}\left[0.494 - \frac{87}{270}\cdot 0.354 - \frac{183}{270}\cdot 0.496\right]$$

$$\approx 0.044$$

Mean Decrease in Impurity (MDI)

$$\Delta I_q = \left(\frac{n}{N}\right)\left[Gini_n - \sum\left(\frac{m}{n}\right) Gini_m\right]$$

Step 1: Calculate the mean decrease in impurity for each node q in the decision tree.



Impurity Decrease

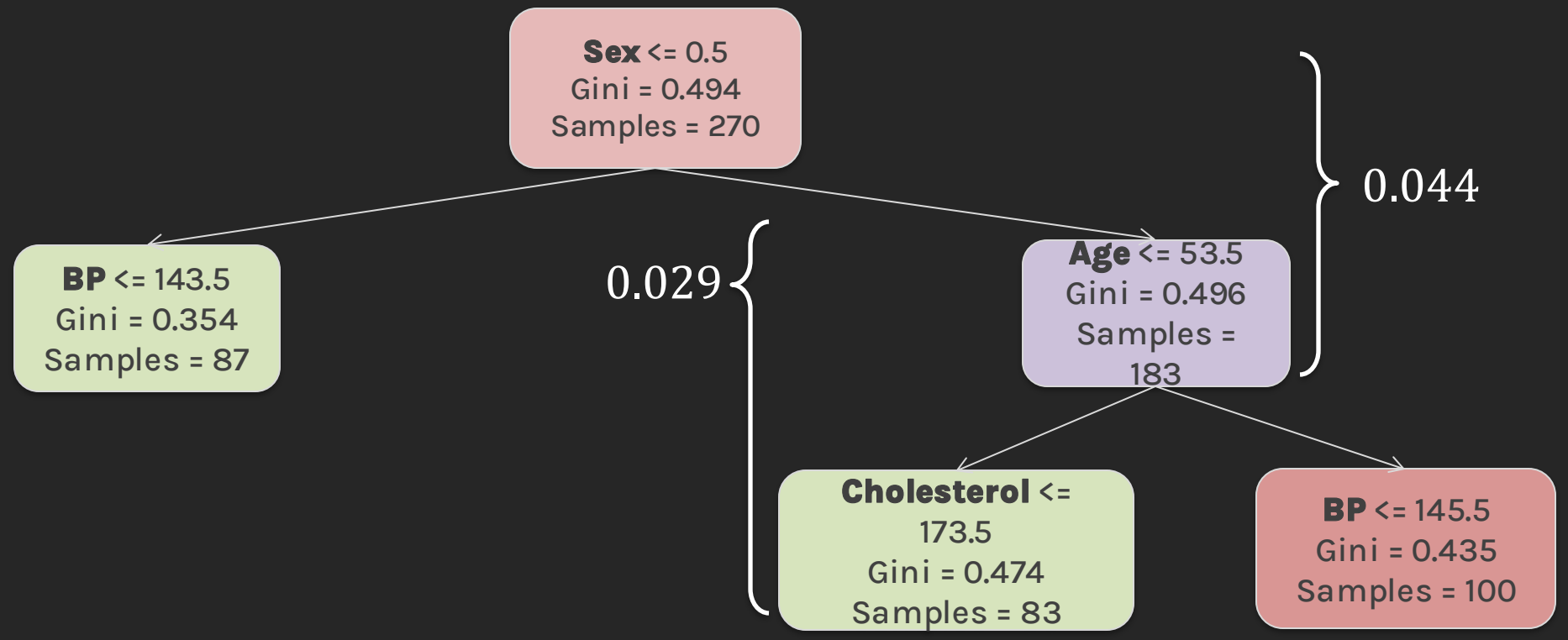
$$= \frac{183}{270}\left[0.496 - \frac{83}{183}\cdot 0.474 - \frac{100}{183}\cdot 0.435\right]$$

$$\approx 0.029$$

Mean Decrease in Impurity (MDI)

$$\Delta I_q = \left(\frac{n}{N}\right)\left[Gini_n - \sum\left(\frac{m}{n}\right) Gini_m\right]$$

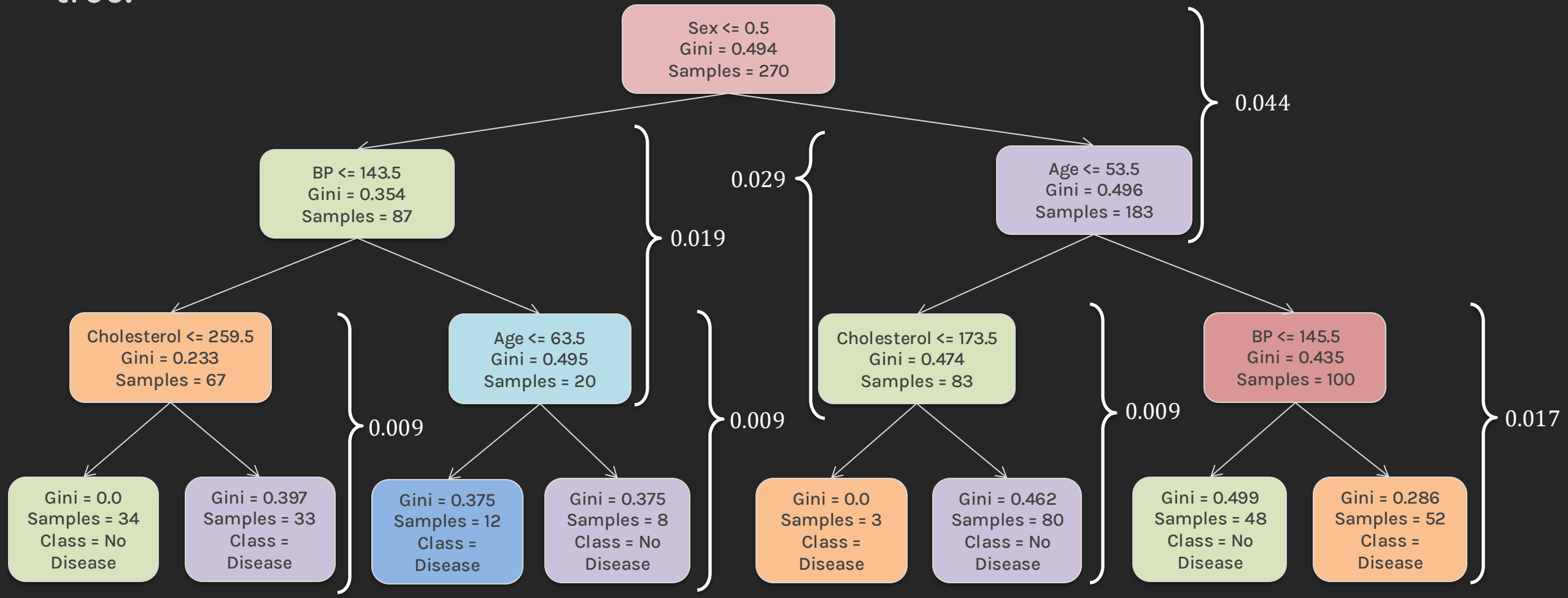
Step 1: Calculate the mean decrease in impurity for each node q in the decision tree.



Mean Decrease in Impurity (MDI)

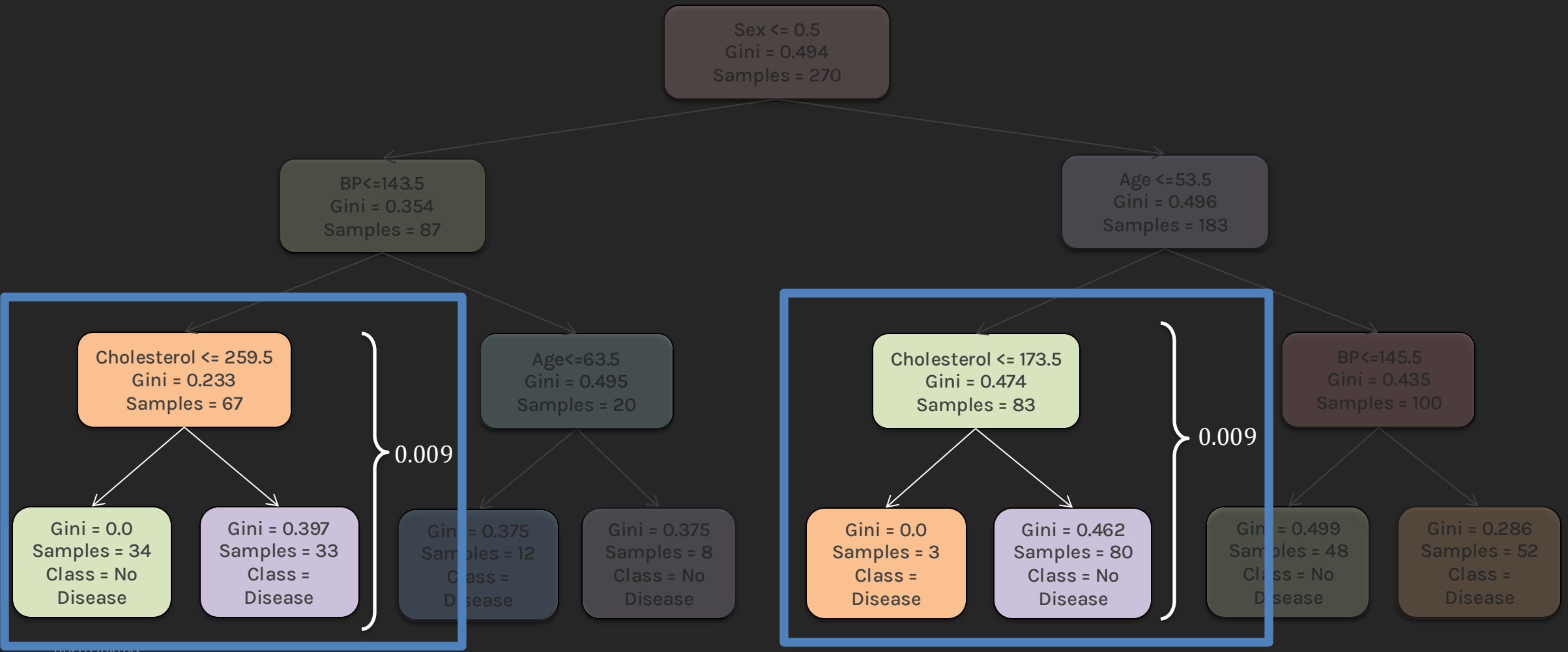
$$\Delta I_q = \left(\frac{n}{N}\right)\left[Gini_n - \sum\left(\frac{m}{n}\right) Gini_m\right]$$

Step 1: Calculate the mean decrease in impurity for each node q in the decision tree.



Mean Decrease in Impurity (MDI)

Let us try to compute the importance of the feature **cholesterol**:



Mean Decrease in Impurity (MDI)

Step 2: Calculate the importance of the feature by **summing up** the impurity decrease calculated in step 1 for each node in which that feature occurs.

$$\text{Feature Importance}_{cholesterol} = \sum_{n \in nodes\ split\ on\ cholesterol} Impurity\ Decrease_n$$

$$= 0.009 + 0.009 = 0.018$$

Mean Decrease in Impurity (MDI)

Step 3: **Normalize** this value between 0 and 1 by dividing by the sum of all feature importance values.

This ensures the sum of all feature importance in a decision tree adds up to 1.

$$\text{Norm Feature Importance}_{cholesterol} = \frac{\text{Feature Importance}_{cholesterol}}{\sum_{j \in all\ features} \text{Feature Importance}_j}$$

Mean Decrease in Impurity (MDI)

Step 4: To calculate the feature importance at the Random Forest or Bagging level, we **average** the normalized feature importance of the given predictor over **all the trees**.

$$\text{Norm RF/Bagging Feature Importance}_{cholesterol} = \frac{\sum_{t \in all\ trees} \text{Norm Feature Importance}}{Total\ number\ of\ trees}$$

Summary: Mean Decrease in Impurity (MDI)

For each feature j in the dataset:

Step 1: Calculate the **mean decrease in impurity** for **each node** n in the decision tree t.

Step 2: Calculate the **feature importance** by **summing up** the impurity decrease calculated in step 1 for each node n in which that feature j occurs.

$$F_j^{(t)} = \sum_{n \in nodes_j} I_n^{(t)}$$

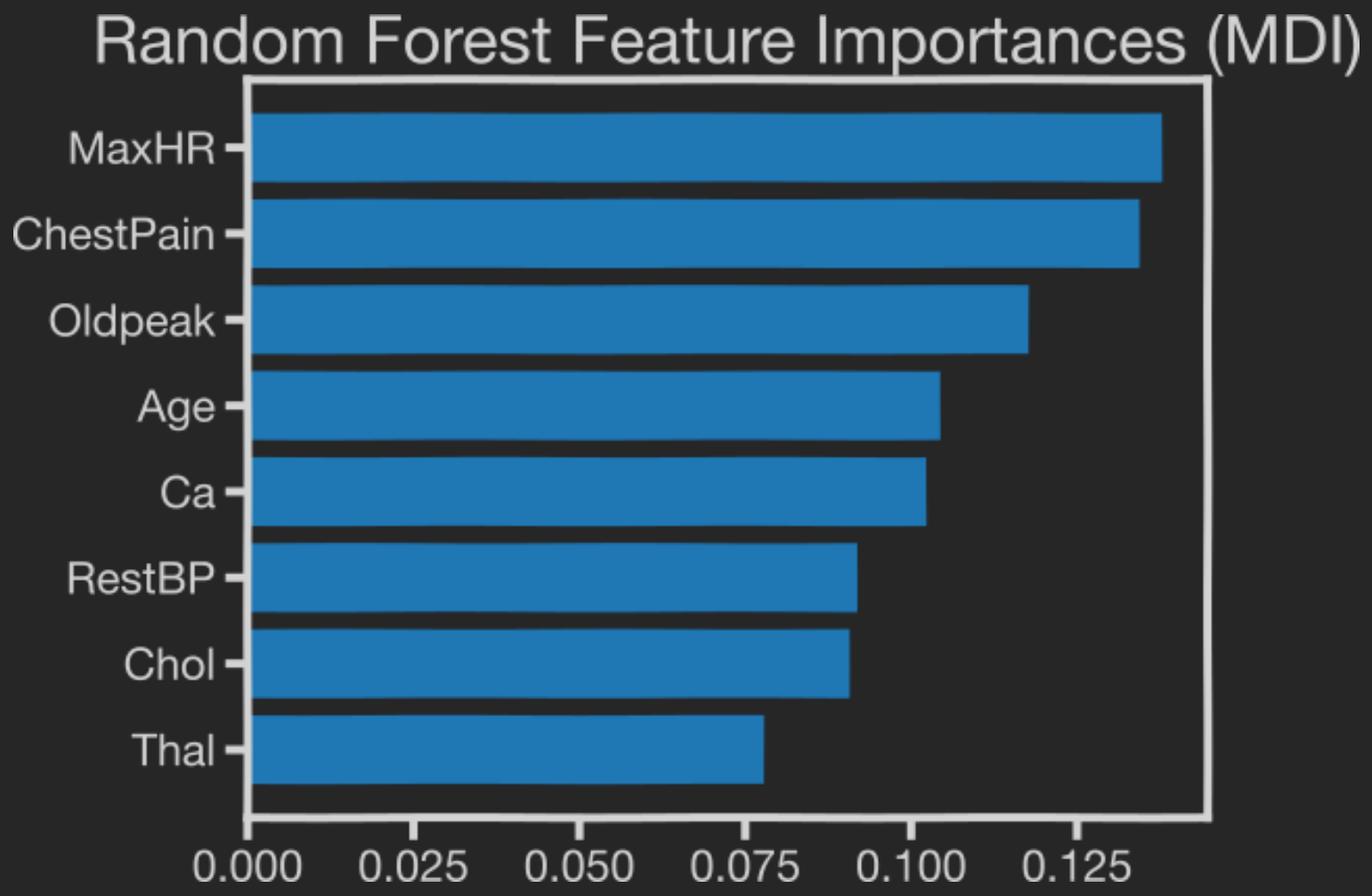
Step 3: **Normalize** this value between 0 and 1 by dividing by the sum of all feature importance values.

$$\hat{F}_j^{(t)} = \frac{F_j^{(t)}}{\sum_i F_i^{(t)}}$$

Step 4: At the Random Forest level, **average** over all the T trees.

$$\mathcal{F}_j = \frac{\sum_t \hat{F}_j^{(t)}}{T}$$

Summary: Mean Decrease in Impurity (MDI)



Variable Importance for RF (and Bagging)

1. Mean Decrease in Impurity.
2. **Permutation importance.**
3. LIME, SHAP values [lab and reading]

Permutation Importance

Consider the following dataset:

Height (cm)	Weight (kg)	…	Fitness Level (1 – 5)
150	65	…	2
140	50	…	3
…	…	…	…
170	70	…	4
160	80	…	1

Step 1: Record the validation/OOB accuracy of RF model:
Accuracy = 0.88

Permutation Importance

Step 1: Record the validation/OOB accuracy of RF model:
Accuracy = 0.88

Height (cm)	Weight (kg)	…	Fitness Level (1 – 5)
150	65	…	2
140	50	…	3
…	…	…	…
170	70	…	4
160	80	…	1

Step 2: Randomly permute the data for column *j* in the validation/OOB set.

Permutation Importance

Step 1: Record the validation/OOB accuracy of RF model::
Accuracy = 0.88

Height (cm)	Weight (kg)	…	Fitness Level (1 – 5)
150	70	…	2
140	65	…	3
…	…	…	…
170	80	…	4
160	50	…	1

Step 2: Randomly permute the data for column j in the validation/OOB set.

Permutation Importance

Step 1: Record the validation/OOB accuracy of RF model:
Accuracy = 0.88

Height (cm)	Weight (kg)	…	Fitness Level (1 – 5)
150	70	…	2
140	65	…	3
…	…	…	…
170	80	…	4
160	50	…	1

Step 2: **Randomly permute** the data for the column j (in this case Weight).

Record the validation/OOB accuracy of the RF model on the modified dataset:
Permuted Accuracy = 0.87

Permutation Importance

Step 1: Record the validation/OOB accuracy of RF model:
Accuracy = 0.88

Step 2: **Randomly permute** the data for the column j (in this case Weight).

Record the validation/OOB accuracy of the RF model on the modified dataset:
Permuted Accuracy = 0.87

Step 3: Repeat the previous step with K number of times and average all the accuracies.

Assume we permute the feature 3 times, we get:

Permuted Accuracy$_1$ = 0.82

Permuted Accuracy$_2$ = 0.87

Permuted Accuracy$_3$ = 0.86

$\longrightarrow$ **Avg Permuted Accuracy=** $\frac{0.82+0.87+0.86}{3}$ **= 0.85**

Permutation Importance

Step 1: Record the validation/OOB accuracy of RF model:
Accuracy = 0.88

Step 2: **Randomly permute** the data for the column j (in this case Weight).

Record the validation/OOB accuracy of the RF model on the modified dataset:
Permuted Accuracy = 0.87

Step 3: Repeat the previous step with K number of times and average all the accuracies:
Avg Permuted Accuracy = 0.85

Step 4: Calculate the difference between unpermuted and average permuted accuracy to get the importance of the feature in the random forest:
Difference = 0.88 - 0.85 = 0.03

Summary: Permutation Importance

For each feature j in the dataset:

Step 1: Record the validation/OOB (unpermuted) accuracy of RF model: s .

Step 2: For each repetition k in 1 .... K:

Randomly permute the data for the column j. Record the validation/OOB accuracy $s_{k,j}$ of the RF model on the modified dataset.

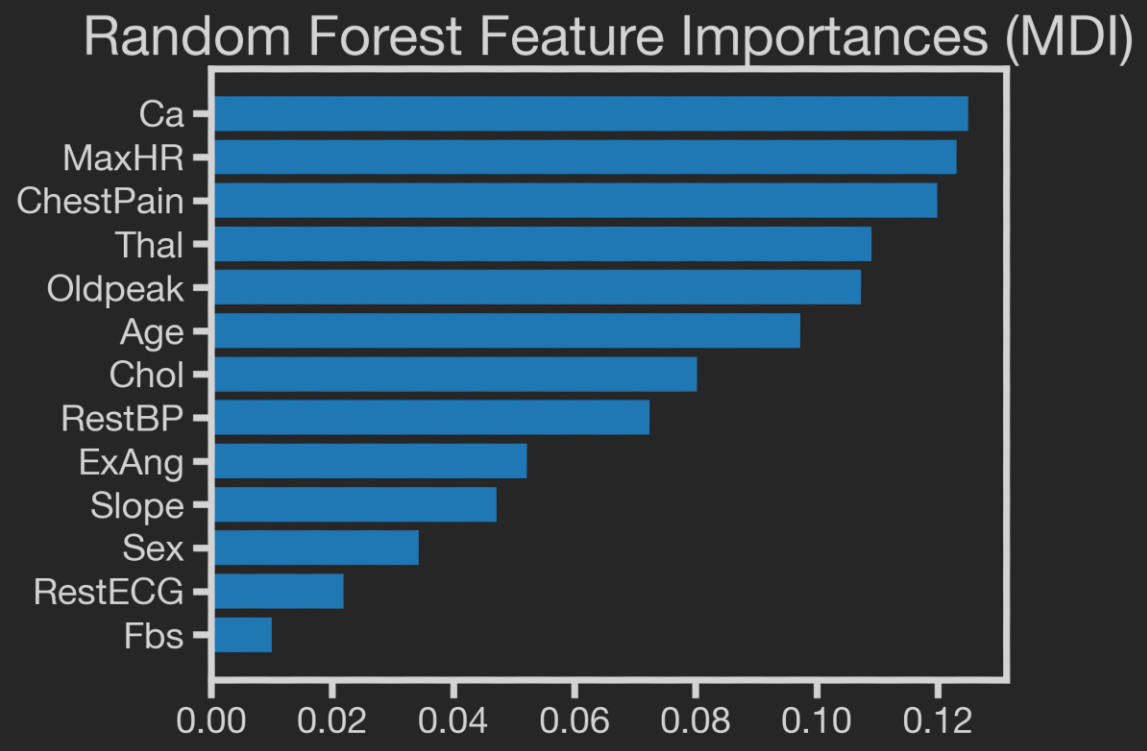
Step 3: Compute the average accuracy from all the permuted datasets

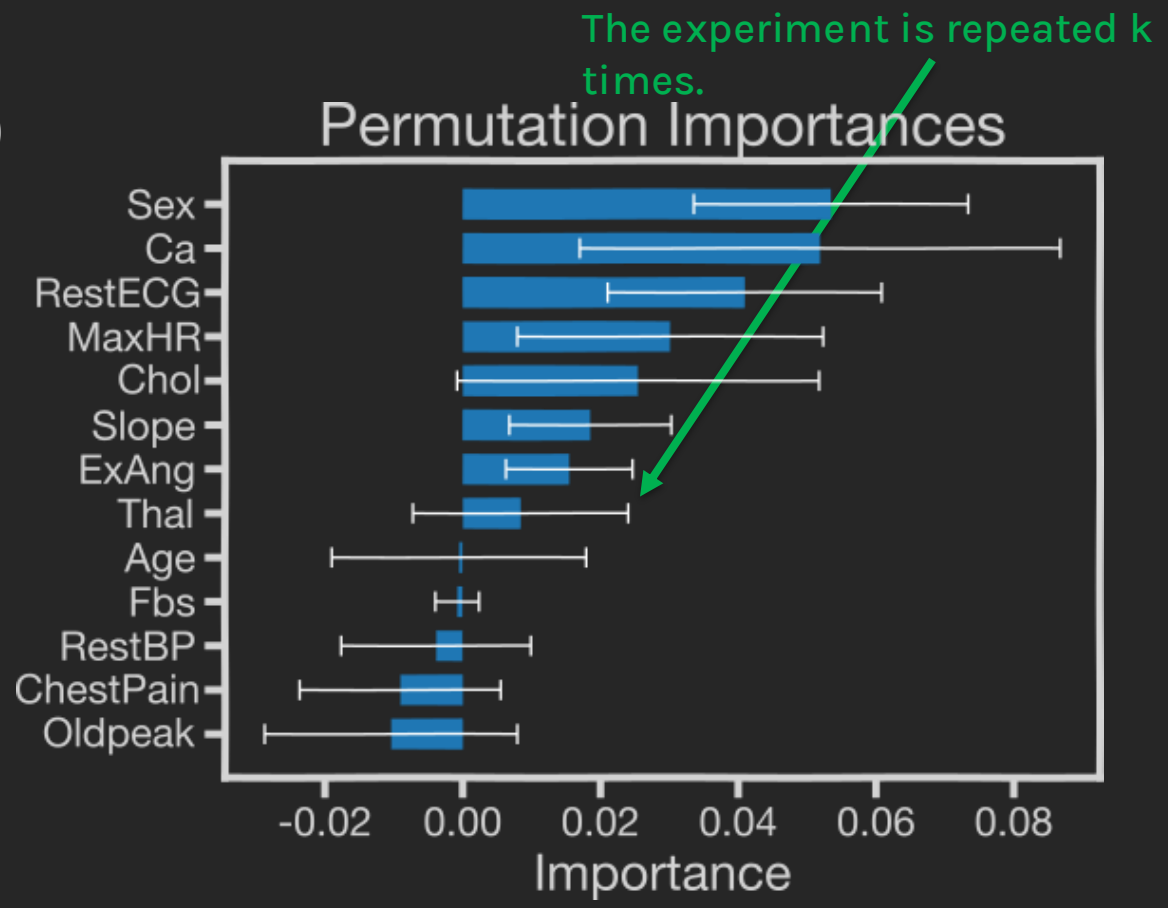
$$s_j = \frac{1}{K}\sum_{k=1}^{K} s_{k,j}$$

Step 4: Calculate the difference between unpermuted and average permuted accuracy to get the importance of the feature in the random forest.

$$\text{RF Feature Importance}_j = s - s_j$$

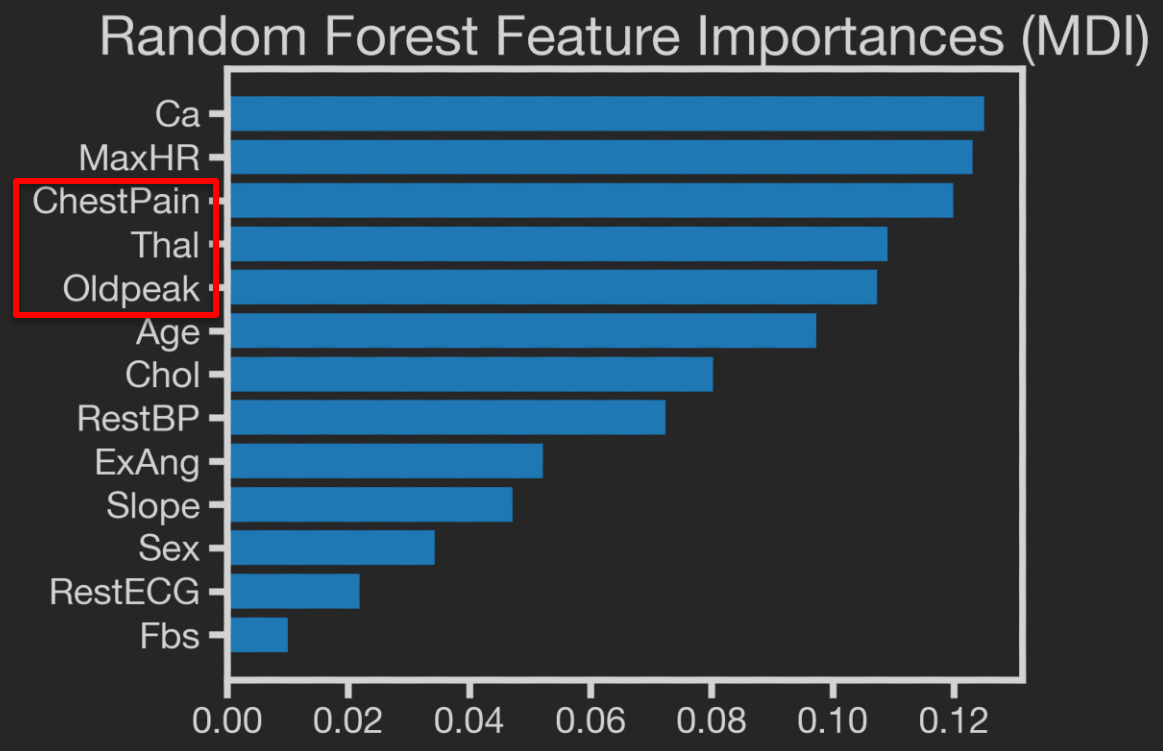
MDI vs Permutation Importance

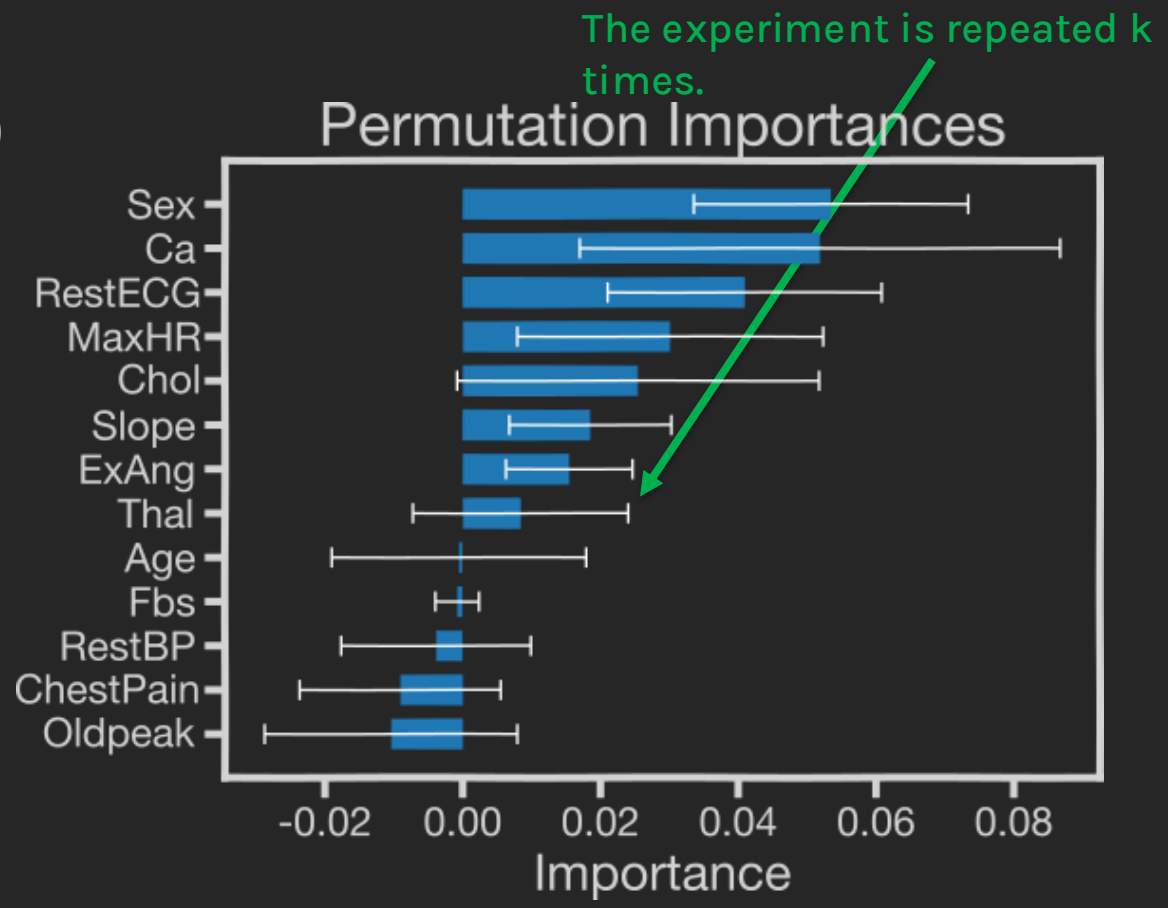




Features like ***ChestPain***, ***OldPeak***, ***Thal*** are ranked most important in MDI importance plot, but they are ranked low in permutation importance plot.

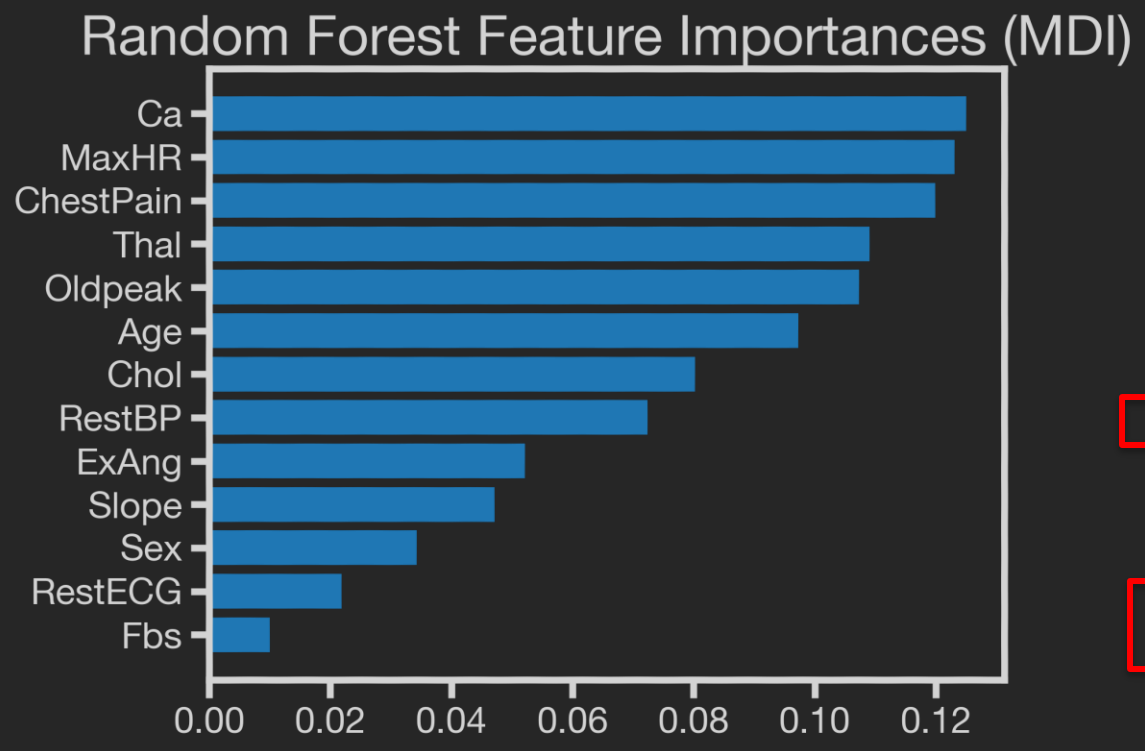
MDI vs Permutation Importance

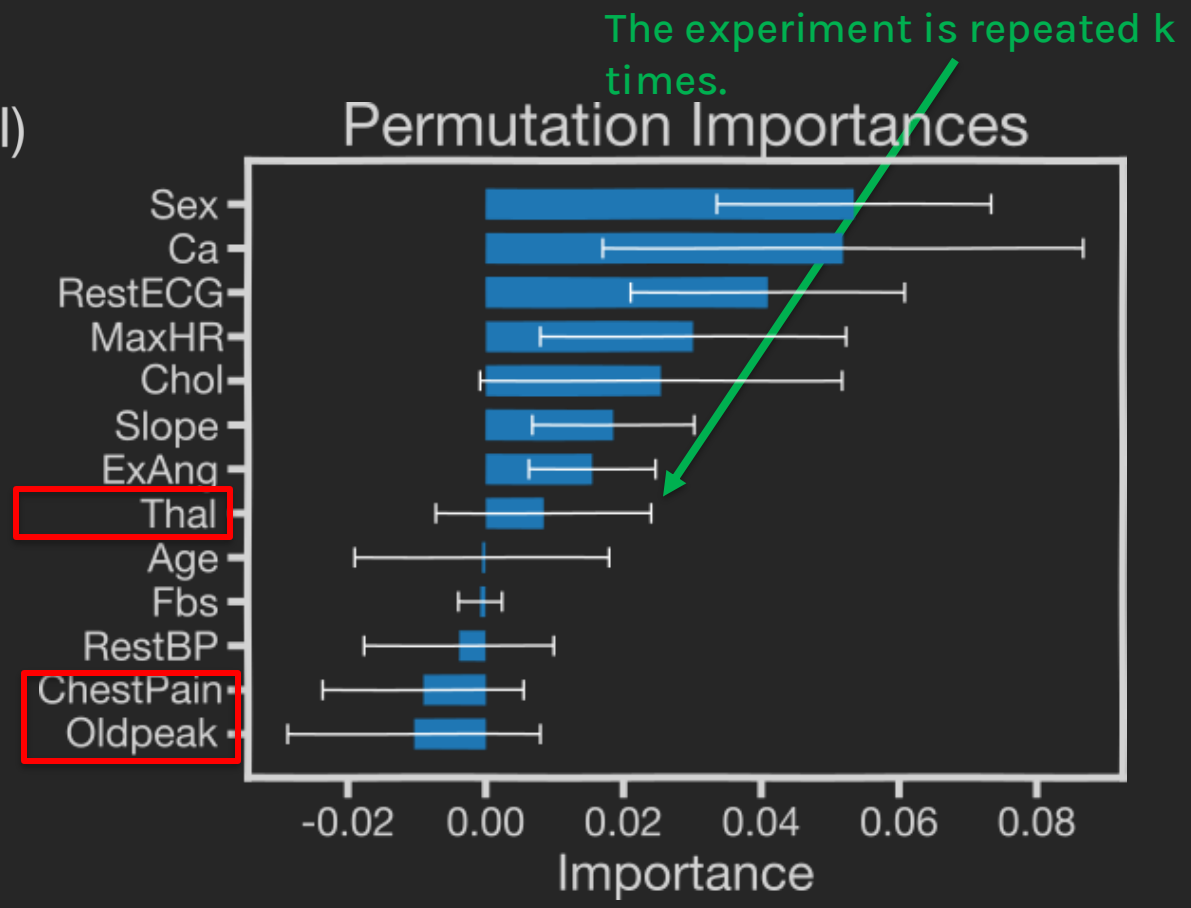




Features like ***ChestPain***, ***OldPeak***, ***Thal*** are ranked most important in MDI importance plot, but they are ranked low in permutation importance plot.

MDI vs Permutation Importance



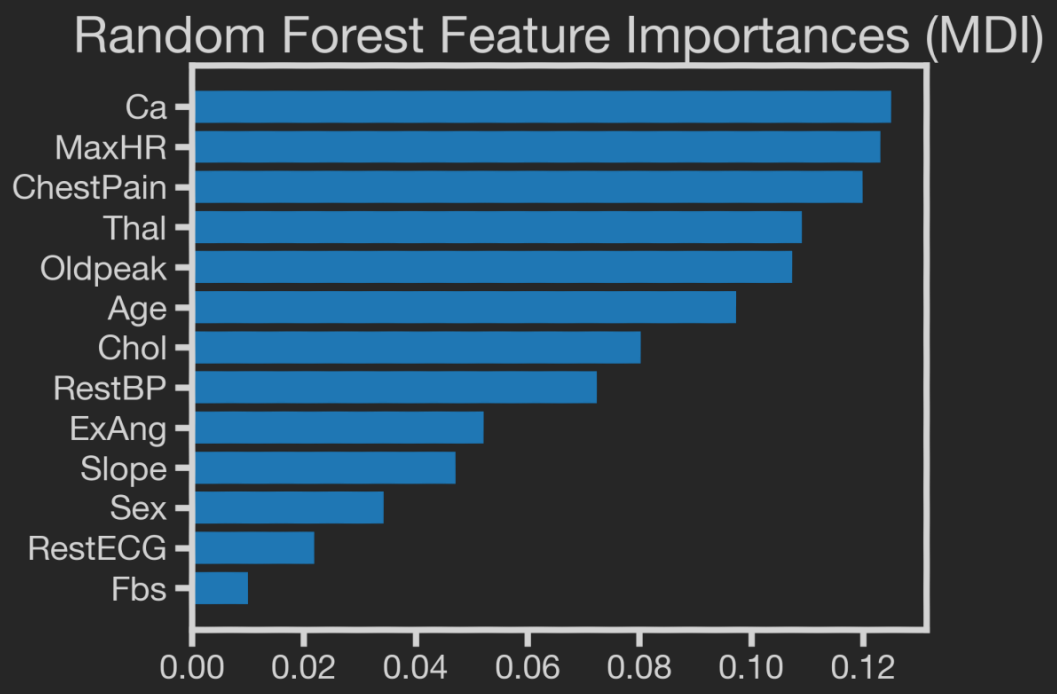


Features like ***ChestPain***, ***OldPeak***, ***Thal*** are ranked most important in MDI importance plot, but they are ranked low in permutation importance plot.

MDI vs Permutation Importance

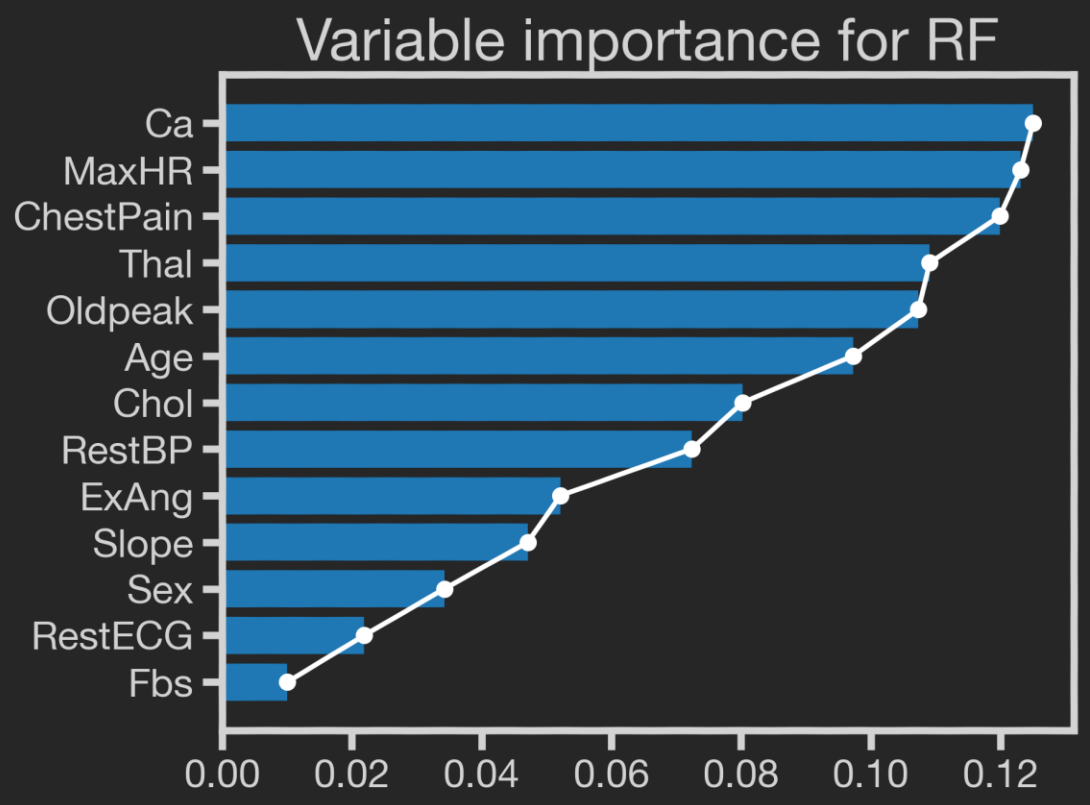
- The biggest advantage of the MDI is **speed of computation**. All needed values are computed during the Random Forest training.

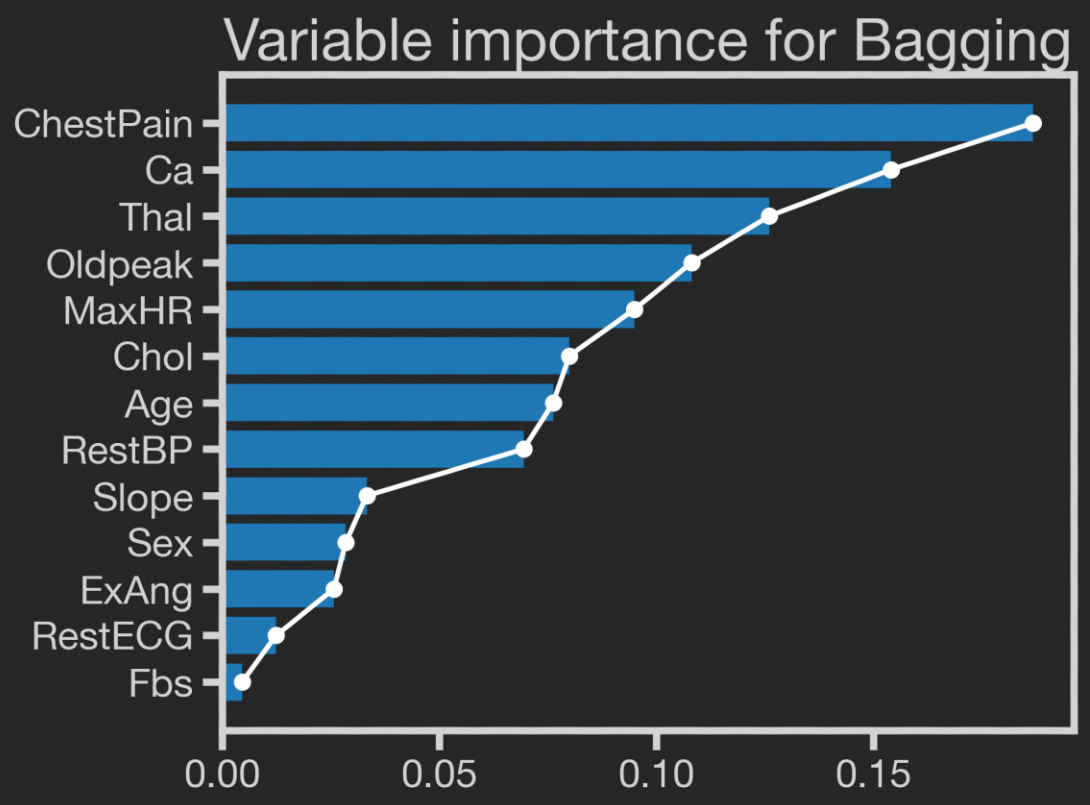
- The drawbacks of the method is its tendency to prefer (select as important) numerical features and categorical features with **high cardinality**. In the example shown, *Max HR* is selected as an important feature because of the high cardinality.



Variable Importance for bagging vs RF

Variable importance for RF is smoother than that for bagging due to randomness introduced by selecting a subset of predictors to choose from.



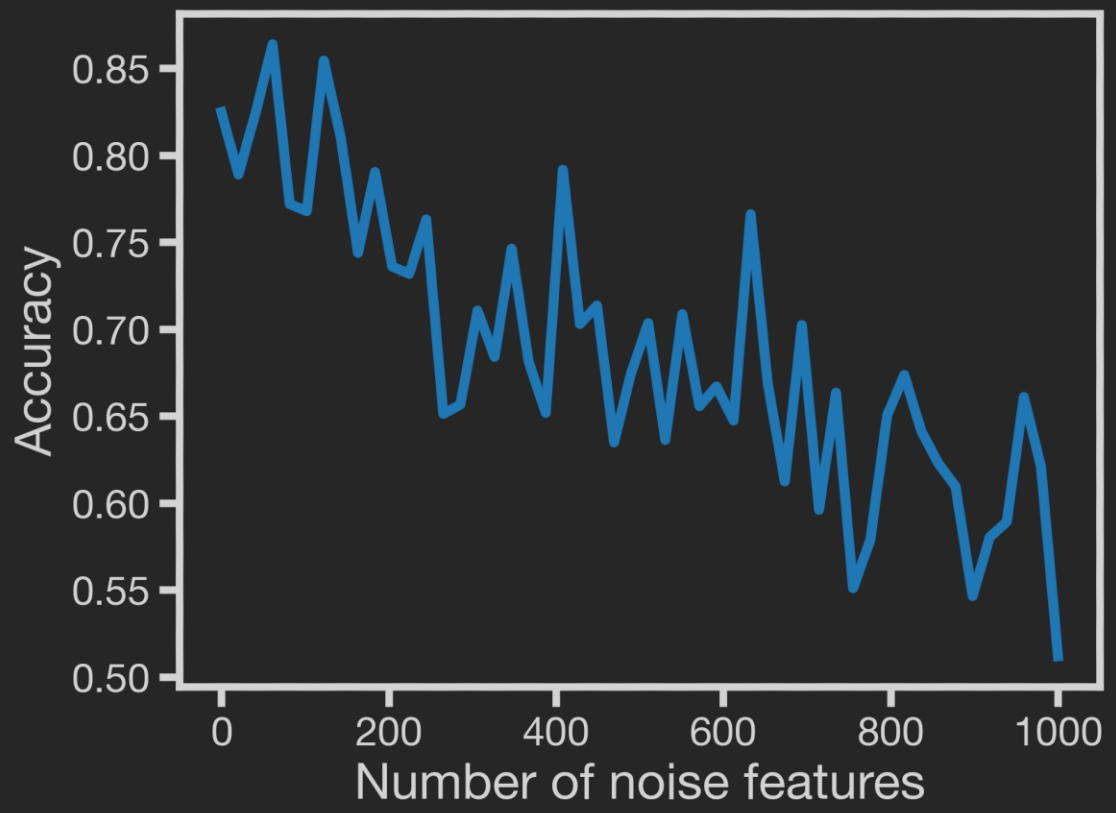


Final Thoughts on Random Forests

When the number of predictors is large, but the number of relevant predictors is small, random forests can perform poorly.

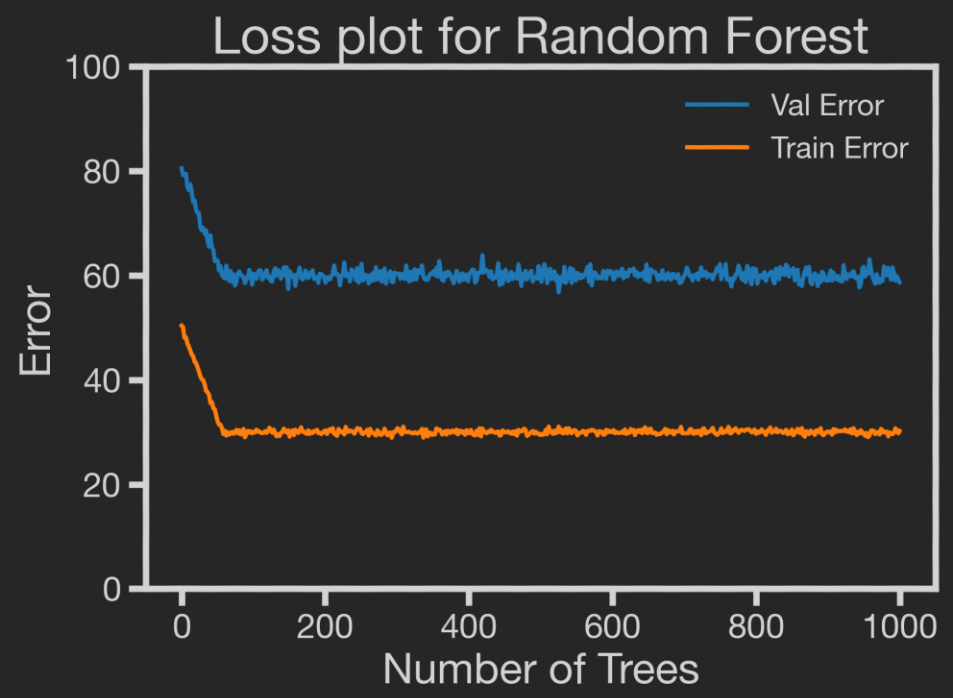
Question: Why?

In each split, the chances of selecting a relevant predictor will be low and hence most trees in the ensemble will be weak models.



Final Thoughts on Random Forests

Increasing the number of trees in the ensemble generally does not increase the risk of overfitting.



By decomposing the generalization error in terms of bias and variance, we see that increasing the number of trees produces a model that is at least as robust as a single tree.

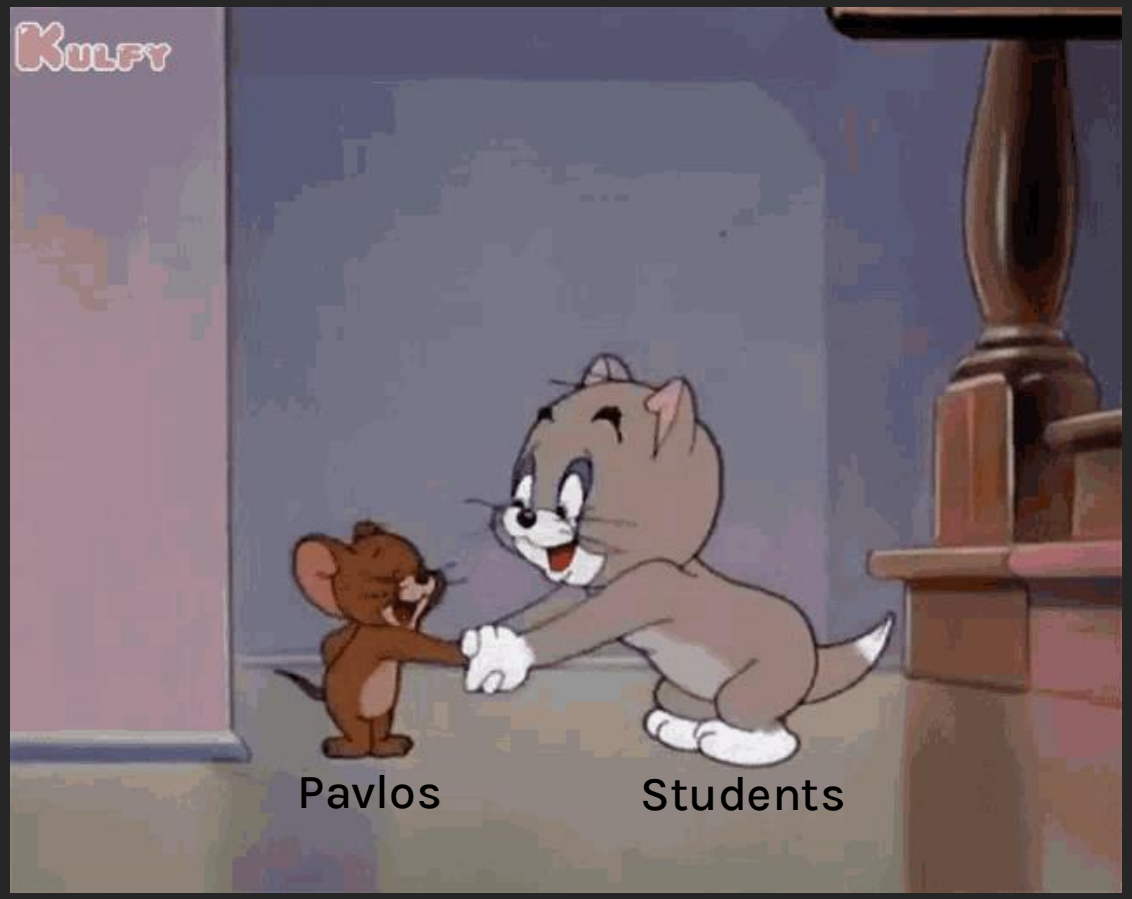
Final Thoughts on Random Forests

Random Forest Classifier (and bagging) can return probabilities.

Question: How?

The predicted class probabilities of an input sample is computed as the mean predicted class probabilities of the trees in the forest.

When you finally understand ‘Random Forest’



Thank you!