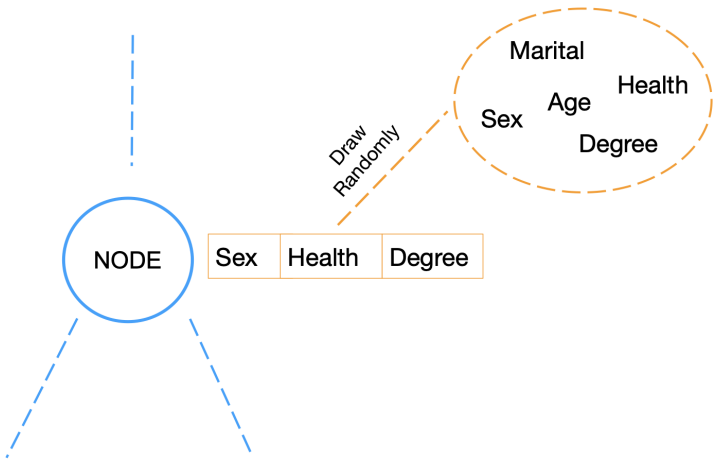
# Introduction to Machine Learning

# Random Forest: In a Nutshell

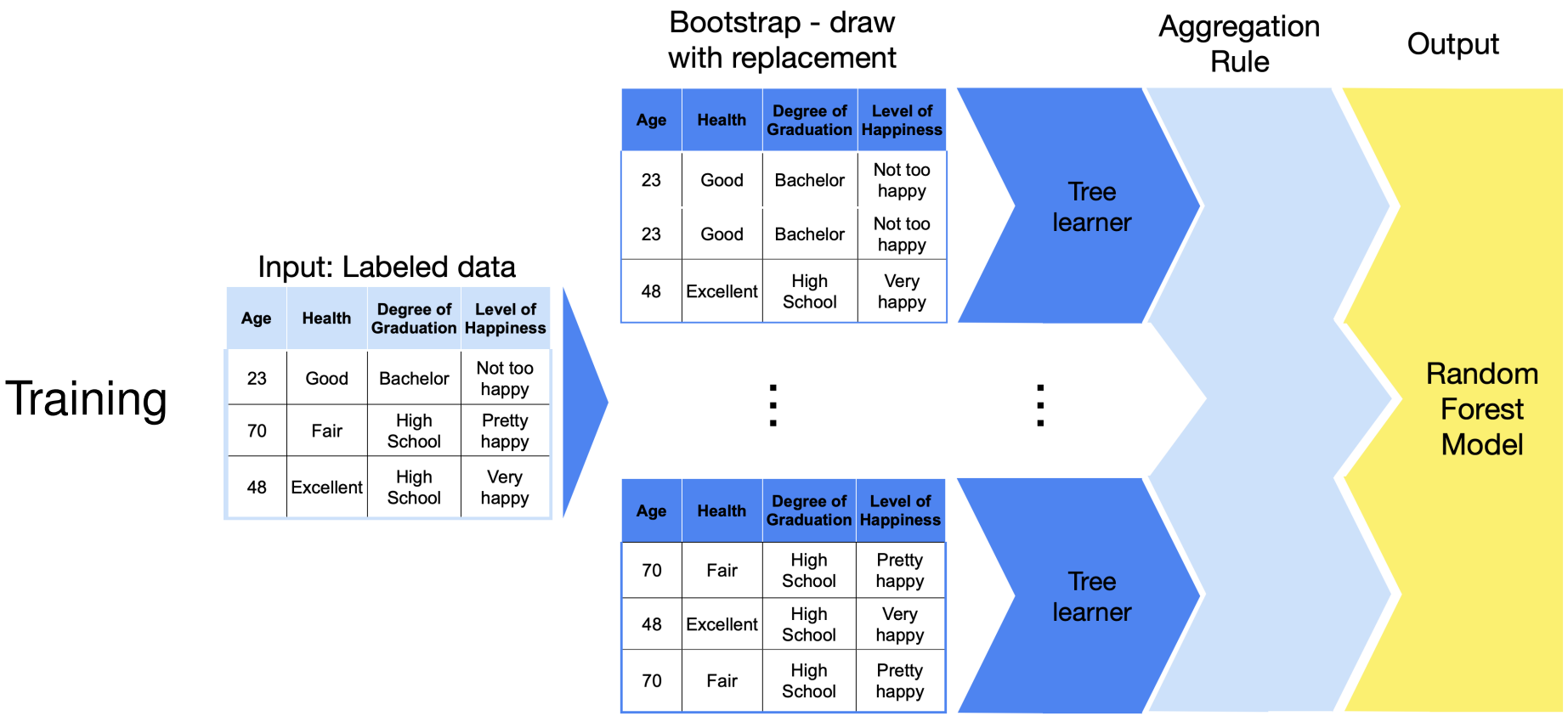



## Learning goals

- Understand basic concept of random forest
- Know basic aggregation rules
- Understand concept of feature importance

# LEARNING AND PREDICTION WITH RF

- Stabilizes tree learner by bagging (bootstrap aggregation)
- Randomizes tree learner and combines models into one meta model
- Can be adapted to learning task, i.e., classification or regression




| Age | Health | Degree of Graduation | Level of Happiness |
|---|---|---|---|
| 23 | Good | Bachelor | Not too happy |
| 70 | Fair | High School | Pretty happy |
| 48 | Excellent | High School | Very happy |

| Age | Health | Degree of Graduation | Level of Happiness |
|---|---|---|---|
| 23 | Good | Bachelor | Not too happy |
| 23 | Good | Bachelor | Not too happy |
| 48 | Excellent | High School | Very happy |

| Age | Health | Degree of Graduation | Level of Happiness |
|---|---|---|---|
| 70 | Fair | High School | Pretty happy |
| 48 | Excellent | High School | Very happy |
| 70 | Fair | High School | Pretty happy |

# LEARNING AND PREDICTION WITH RF

Prediction

Input: Unlabeled data

| Age | Health | Degree of Graduation | Level of Happiness |
|---|---|---|---|
| 41 | Fair | Bachelor | ? |
| 35 | Good | Bachelor | ? |
| 22 | Fair | High School | ? |

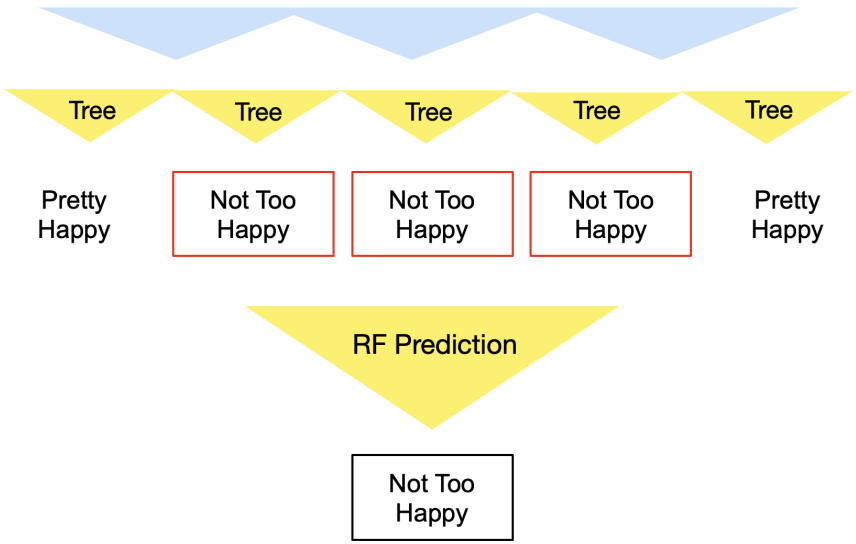

Prediction

| Level of Happiness |
|---|
| Not too happy |
| Pretty happy |
| Not too happy |

# AGGREGATION RULES FOR DIFFERENT TASKS

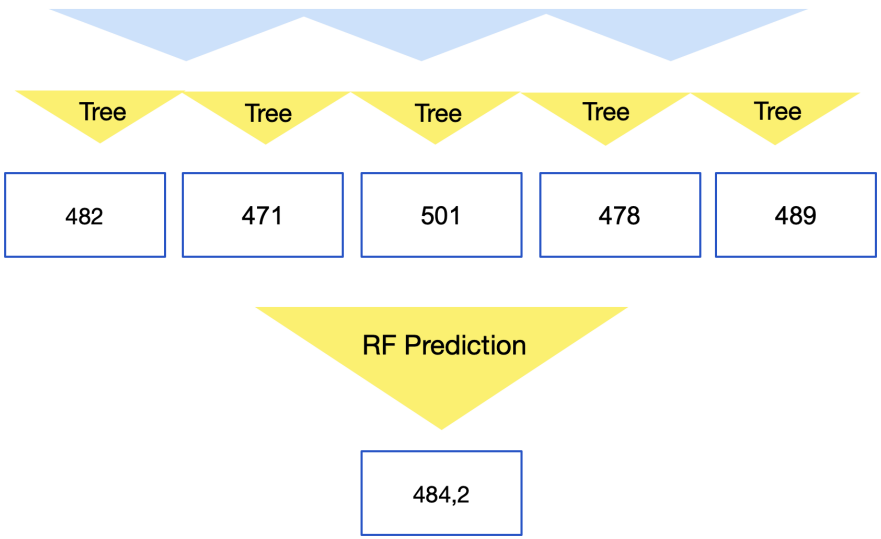
## Classification Task - Majority Vote

| Age | Health | Degree of Graduation | Level of Happiness |
|---|---|---|---|
| 41 | Fair | Bachelor | ? |



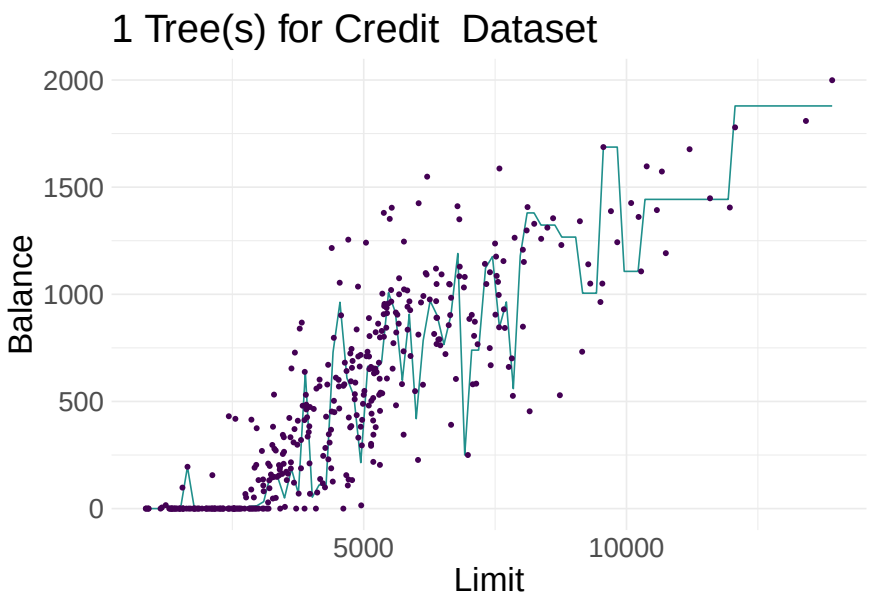

## Regression Task - Averaging

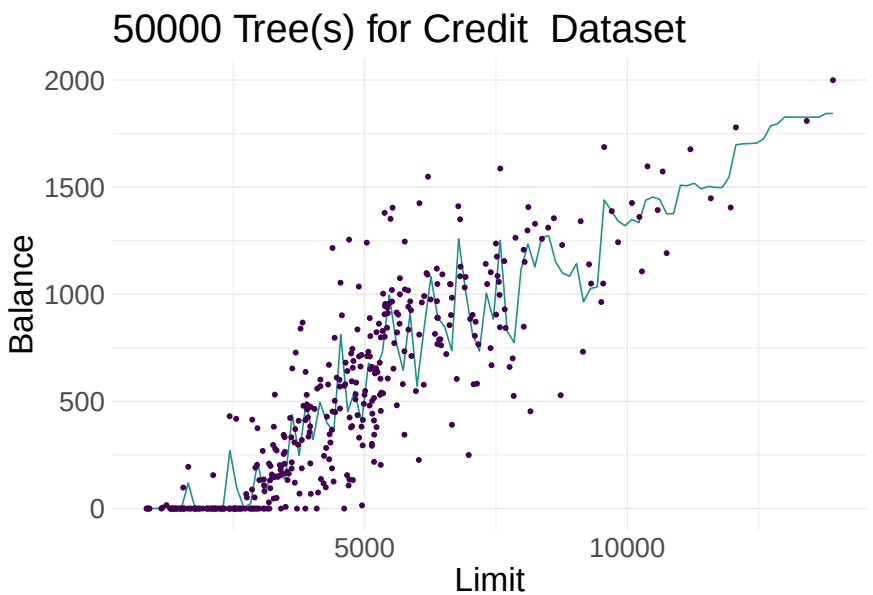
| Rating | Income | Credit Limit | Credit Card Balance |
|---|---|---|---|
| 107 | 32.318 | 4351 | ? |



# PERFORMANCE OF RF

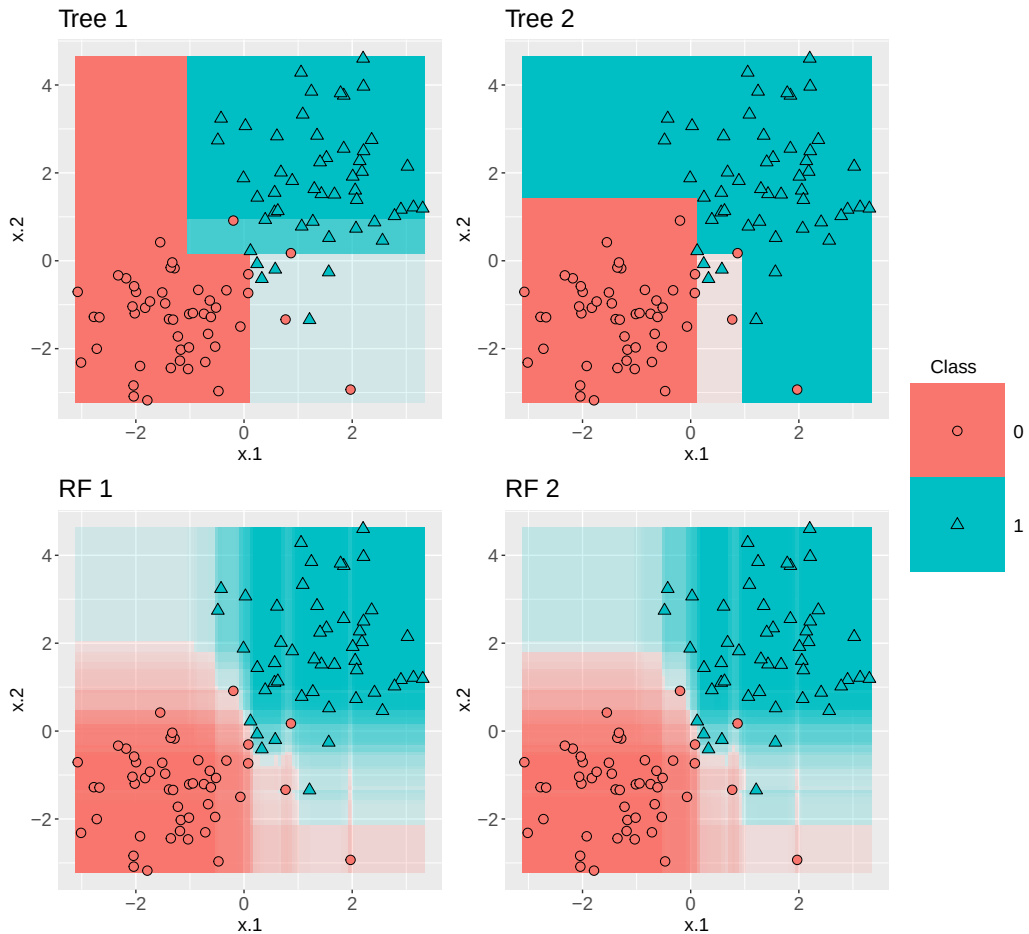
- In general: Increasing the ensemble size stabilizes the predictions
  - For regression tasks the stabilization is often not sufficient.



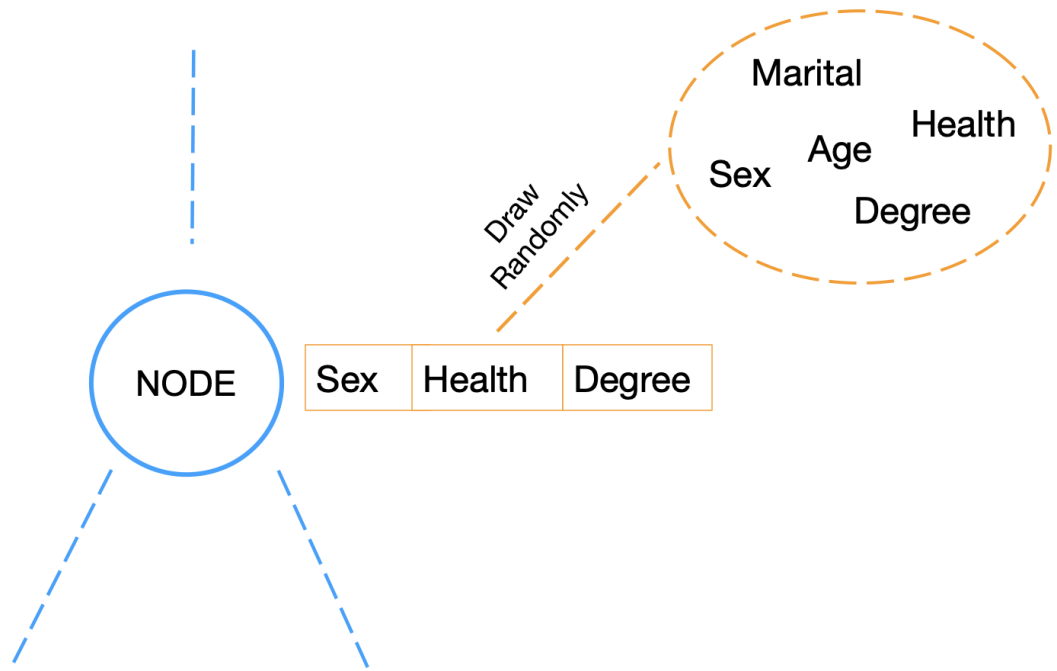
# PERFORMANCE OF RF

- RF performs well for classification tasks:
  - Two different trees $\rightarrow$Quite different decision regions
  - Two different RFs $\rightarrow$Similar decision regions



# PERFORMANCE OF RF

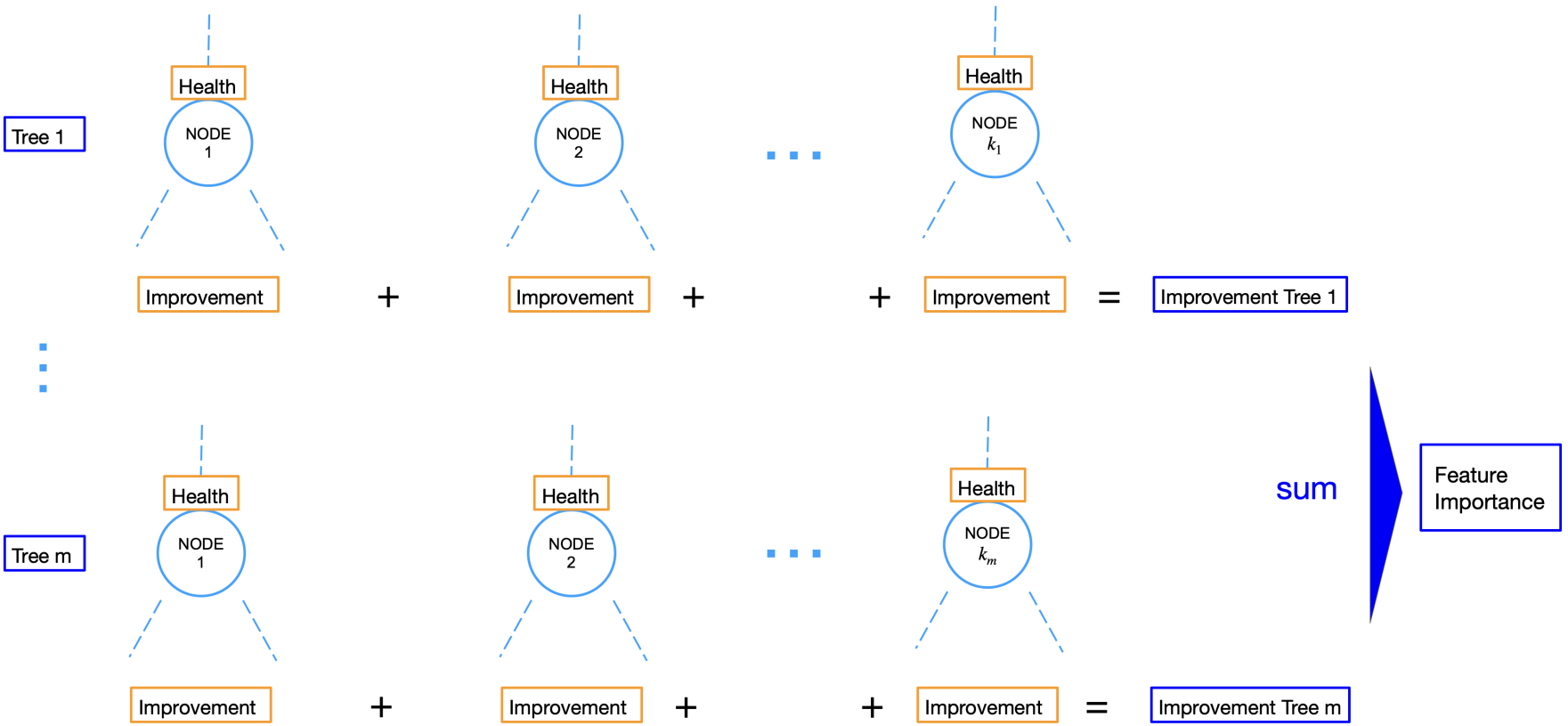
- Trees should be decorrelated, i.e., make mistakes in different directions
- Avoid correlation by
  - Bootstrap sampling
  - Randomized splits. In each node of each tree, consider different features for splitting:



# FEATURE IMPORTANCE

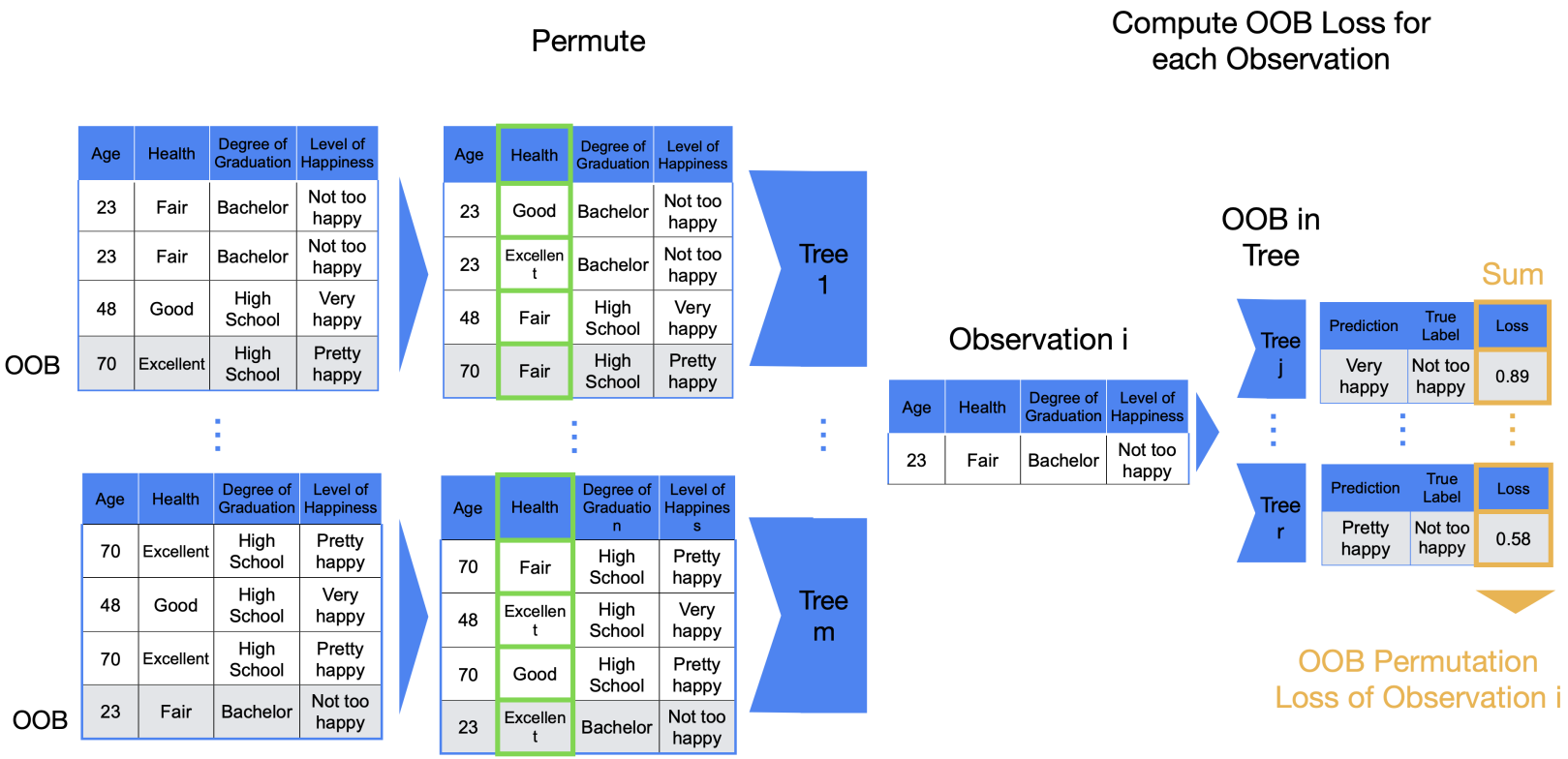
Several options, e.g., measure contribution of feature to model:

- Measure based on improvement in splitting criterion
- E.g. Feature importance of 'Health', search all nodes with 'Health' as splitting variable:

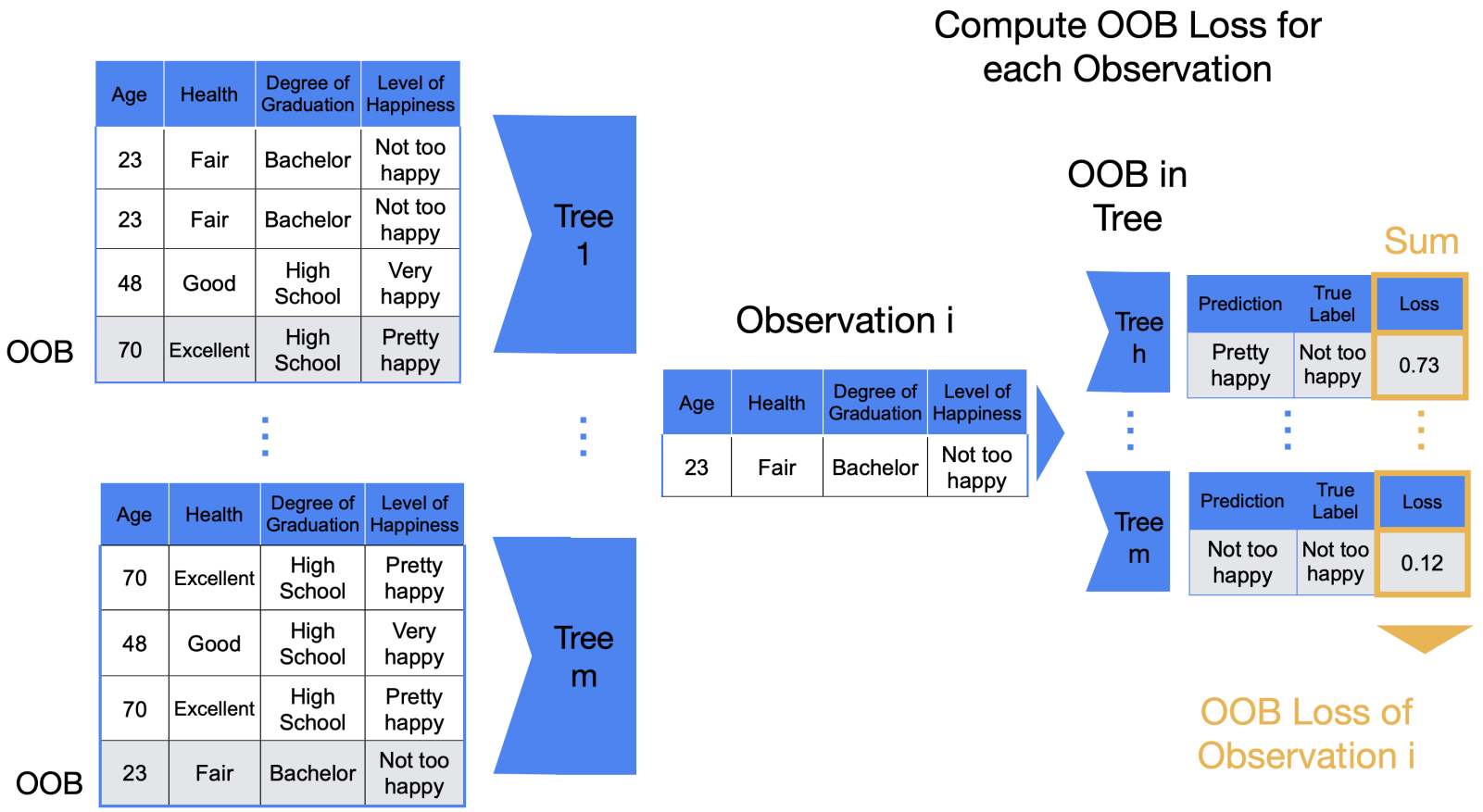


# FEATURE IMPORTANCE

- Measure based on OOB Loss



# FEATURE IMPORTANCE




**Feature Importance 'Health'** = **OOB Permutation Loss of all observations** - **OOB Loss of all observations**