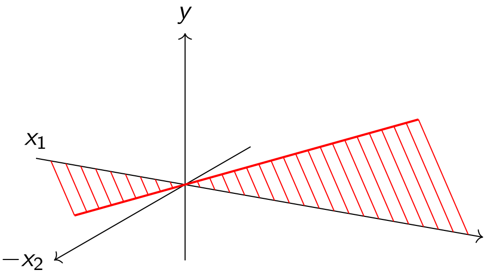
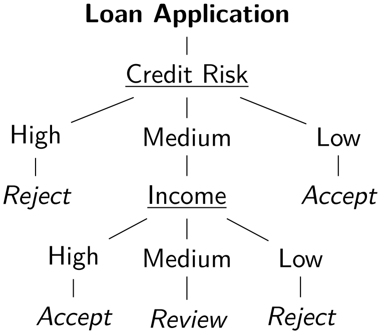
4. Interpretable-by-design models

Introduction

- for some models, **interpretability not an issue**
- **Examples:**
  - linear models
  - decision trees

$$y = 0.5 \cdot x_1 - 0.2 \cdot x_2$$





Loan Application

Credit Risk

High, Medium, Low

Reject, Income, *Accept*

High, Medium, Low

Accept, *Review*, *Reject*

4.1. Linear models

Linear models: quick recap

- **Linear models:** output depends linearly on each feature
- mathematically, in the *regression* setting:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_d x_d \,,$$

- given training data $(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(n)})$, model is fitted by *ordinary least squares*

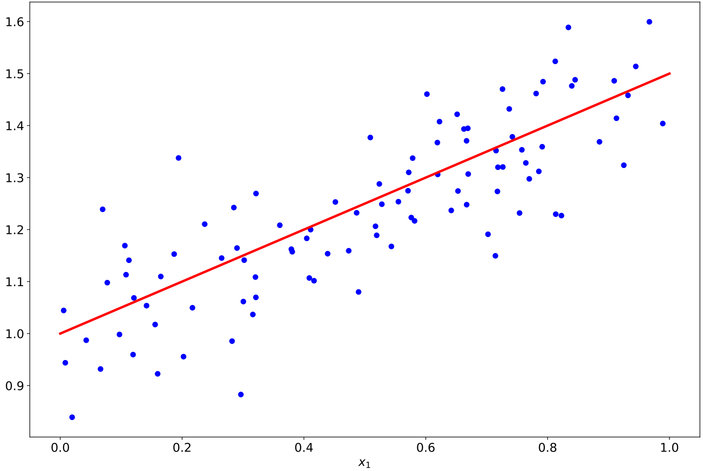
$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(Y^{(i)} - \beta_0 - \sum_{j=1}^{d} \beta_j X_j^{(i)} \right)^2 \right\}$$

- **Intuition:** minimize the sum of squares of prediction errors
- usual ways to control the size / number of non-zeros coefficients: ridge[15] and LASSO[16]

[15] Hoerl and Kennard, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics, 1970

[16] Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society Series B: Statistical Methodology, 1996

Linear regression: example



- **Figure:** ordinary linear regression (in 1D)

Linear models: quick recap

- **Ridge regression:** adding a L^2 penalty in the optimization:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(Y^{(i)} - \beta_0 - \sum_{j=1}^{d} \beta_j X_j^{(i)} \right)^2 + \lambda \sum_{j=1}^{d} \beta_j^2 \right\} ,$$

  where $\lambda > 0$ is a hyperparameter

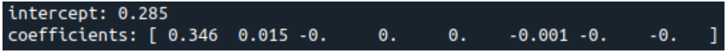
- **Least Absolute Shrinkage and Selection Operator (LASSO):** adding a L^1 penalty:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(Y^{(i)} - \beta_0 - \sum_{j=1}^{d} \beta_j X_j^{(i)} \right)^2 + \lambda \sum_{j=1}^{d} |\beta_j| \right\} ,$$

  where $\lambda > 0$ is a hyperparameter

Linear models: interpretability

- **Why are these models interpretable?** increasing x_j by one unit increases $f(x)$ by β_j
- let us look at a concrete example: regression task on the California housing dataset[17]
- we run LASSO on a train set (75% of the data)
- RMSE on the test is 0.83 (not too bad!)
- we can read the coefficients:



```
intercept: 0.285
coefficients: [ 0.346  0.015 -0.     0.     0.    -0.001 -0.    -0.   ]
```

- **Mathematically:** our model is given by

$$f(x) = 0.285 + 0.346 \cdot x_{\text{MedInc}} + 0.015 \cdot x_{\text{HouseAge}} - 0.001 \cdot x_{\text{AveOccup}} \,.$$

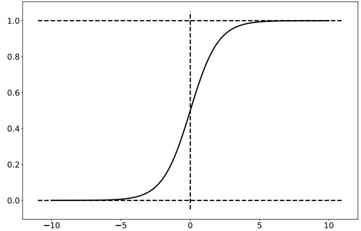
- we can directly read in these coefficients what our model is doing!

[17] Pace and Barry, *Sparse Spatial Autoregressions*, Statistics & Probability Letters, 1997

Logistic regression: quick recap

- **Logistic regression:** linear model in the classification setting
- **Quick reminder:** logistic function is defined as

$$\sigma(z) := \frac{1}{1+\mathrm{e}^{-z}} .$$



- logistic regression models $\mathbb{P}(Y = 1 \,|\, X = x)$ by $\sigma(\beta^\top x)$

Logistic regression: quick recap

- given train data $(X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)})$, we fit the model by solving

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \ell(Y^{(i)}, \sigma(\beta^\top X^{(i)})) \right\} ,$$

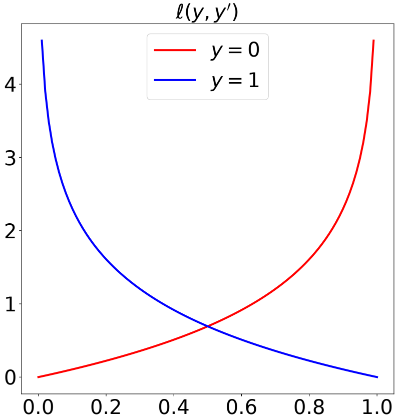
where ℓ is the *cross-entropy loss*

$$\ell(y, y') := -y \log y' - (1 - y) \log(1 - y') \, .$$

- **Intuition:** find coefficients such that prediction score is high when true label is 1
- **In any case:** our model is given by

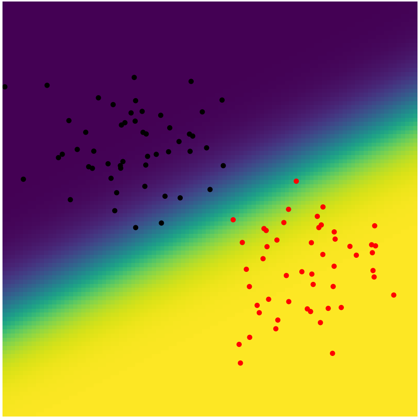
$$f(x) = \frac{1}{1 + \exp\left(-\beta_0 - \beta_1 x_1 - \cdots - \beta_p x_p\right)} \, .$$

Cross-entropy loss



- **Figure:** the cross-entropy loss

Logistic regression: example



- **Figure:** logistic regression in 2D for separable data

Logistic regression: interpretability

- **Why is this interpretable?** σ is monotonous
- thus, if $\beta_j > 0$, increase in x_j means higher score
- more convenient to reason in terms of log-odds:

$$\begin{aligned}\text{log-odds}(x) &= \log \frac{\mathbb{P}\left(Y = 1 \,|\, X = x\right)}{\mathbb{P}\left(Y = 0 \,|\, X = x\right)} \\ &= \log \frac{\frac{1}{1+\exp(-\beta_0-\beta_1 x_1-\cdots-\beta_p x_p)}}{1 - \frac{1}{1+\exp(-\beta_0-\beta_1 x_1-\cdots-\beta_p x_p)}} \\ &= \log \frac{1}{\exp\left(-\beta_0 - \beta_1 x_1 - \cdots - \beta_p x_p\right)} \\ &= \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \,.\end{aligned}$$

- thus **increase of x_j by one unit means increase of log-odds by β_j**

Summary

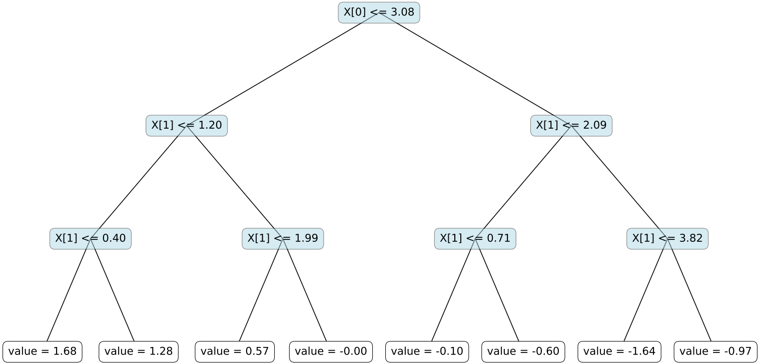
- linear models are light-weight models
- model either the output as a linear transformation of the input...
- ...or probability of belonging to a given class
- **They are interpretable:** directly looking at the coefficients
- **Limitations:**
  - accuracy far from state-of-the-art (model too simple)
  - need meaningful features...
  - and not too many of them

4.2. Decision trees

Decision tree: quick recap

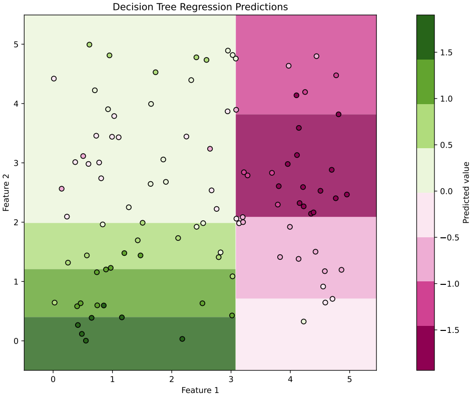
- **What is a decision tree?**
- tree with root $= \mathcal{X}$ and leaves $=$ cells
- iterative binary decisions based on feature values
- node of the tree: "is feature j smaller than x?"
- if yes, go left, if not, go right
- **Can also be visualized as** partition of the input space $\mathcal{X}$
- each query point falls into a cell, constant prediction on each cell
- two different modes:
    - classification → class label → **majority vote**
    - regression → real number → **empirical average**

Decision tree: example



- **Figure:** example of a decision tree for regression in 2D

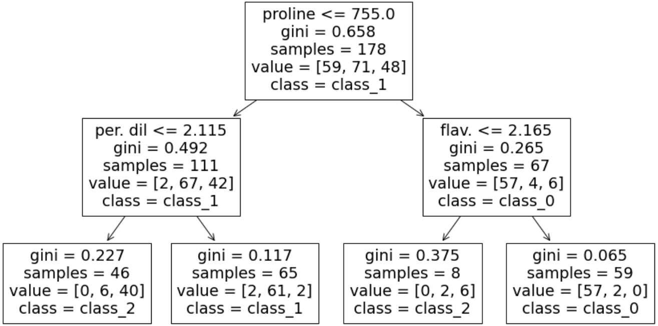
Decision tree: example



- **Figure:** associated partition of the space

Decision trees: interpretability

- **Why is this model interpretable?**
- let us look at a concrete example: the Wine dataset[18]



proline <= 755.0
gini = 0.658
samples = 178
value = [59, 71, 48]
class = class_1

per. dil <= 2.115
gini = 0.492
samples = 111
value = [2, 67, 42]
class = class_1

flav. <= 2.165
gini = 0.265
samples = 67
value = [57, 4, 6]
class = class_0

gini = 0.227
samples = 46
value = [0, 6, 40]
class = class_2

gini = 0.117
samples = 65
value = [2, 61, 2]
class = class_1

gini = 0.375
samples = 8
value = [0, 2, 6]
class = class_2

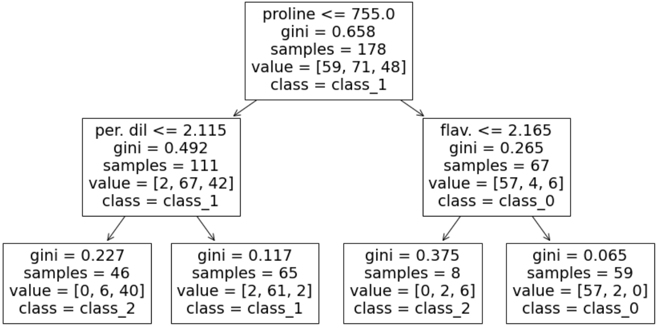
gini = 0.065
samples = 59
value = [57, 2, 0]
class = class_0

[18] Cortez et al., *Modeling wine preferences by data mining from physicochemical properties*, Decision Support Systems, 1998

Decision trees: interpretability

- for a specific example, we can **run down the path** to understand the decision
- we can also infer **global rules**
- for instance, we know that

$$\{\text{proline} \leq 755.0, \text{per. dil} \leq 2.115\} \rightarrow \text{class } 2\,.$$



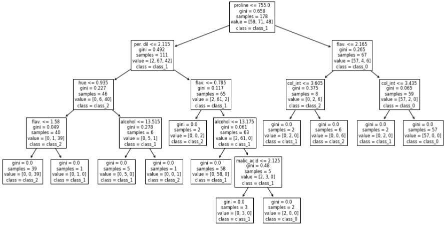
Summary

- decision trees are light-weight models
- recursively splitting the input space according to a numerical criterion
- **They are interpretable:** either tracing down the decision or deducing global rules
- **Limitations:**
  - accuracy far from state-of-the-art (model too simple)
  - interpretability decreases with number of leaves (see next section)

5. Ad-hoc methods

Ad-hoc methods

- even interpretable-by-design models can become un-interpretable
- **Typical scenario:** too many parameters
- **Example:** tree with large width / depth



- we can still leverage the **particular structure of the model**
- → *ad-hoc* importance measures

5.1. Mean decrease impurity

Impurity: classification

- **Key notion for tree construction:** impurity
- **Informally,** quantity measuring how homogeneous a node is
- **Notation:** $(X^{(1)}, Y^{(1)}), (X^{(2)}, Y^{(2)}), \ldots, (X^{(n)}, Y^{(n)})$ training points, $Y^{(i)} \in [K]$
- for each node m and label k, define label *proportion*

$$p_k(m) := \frac{\left|\{i \in [n], X^{(i)} \in m \text{ and } Y^{(i)} = k\}\right|}{N(m)}, \quad \text{where} \quad N(m) := \left|\{i \in [n], X^{(i)} \in m\}\right| .$$

Definition: for a given node m, we define *Gini impurity* as

$$i(m) := \frac{1}{2} \sum_{k=1}^{K} p_k(1 - p_k) .$$

- **Intuition:** "lower is better" (one class in node $\Rightarrow i(m) = 0$)

Impurity decrease: classification

- **Tree construction:** recursively split according to maximal impurity decrease

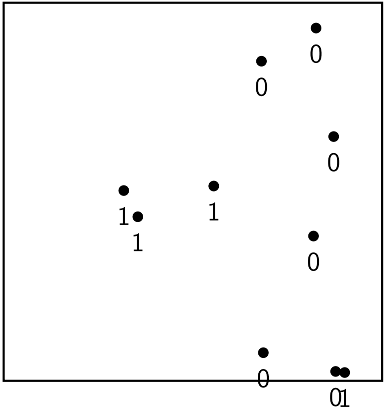
Definition: consider a node m and a possible split along coordinate j with level z. We call m_L and m_R the two new sub-cells (m_L corresponds to $X^{(j)} < z$). The *impurity decrease* is defined as

$$L(j,z) := i(m) - p_L i(m_L) - p_R i(m_R)\,,$$

where p_L (resp. p_R) is the proportion of observations in m falling into m_L (resp. m_R).

- **Intuition:** start with large $i(m)$ and imagine a split producing two "pure" cells
- in that event, $i(m_L) = i(m_R) = 0 \Rightarrow$ large impurity decrease
- in the other direction, "bad" splits produce cells with $i(m_L) \approx i(m_R) \approx i(m)$
- which yields small impurity decrease

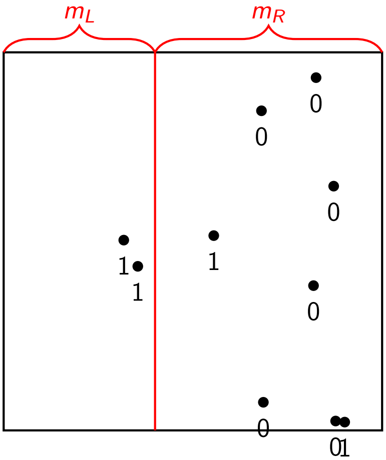
Quick recap: impurity decrease



- **Example:** current cell m has ten points
- $p_0 = 0.6,\ p_1 = 0.4$
- we compute

$$i(m) = \frac{1}{2}\left(0.4 \cdot (1 - 0.4) + 0.6 \cdot (1 - 0.6)\right) = 0.24\,.$$

Quick recap: impurity decrease



- let us look at a first split
- we compute

$$i(m_L) = 0\,,$$

$$i(m_R) = \frac{1}{2}\left(\frac{6}{8}\cdot\left(1-\frac{6}{8}\right)+\frac{2}{8}\cdot\left(1-\frac{2}{8}\right)\right) \approx 0.19$$

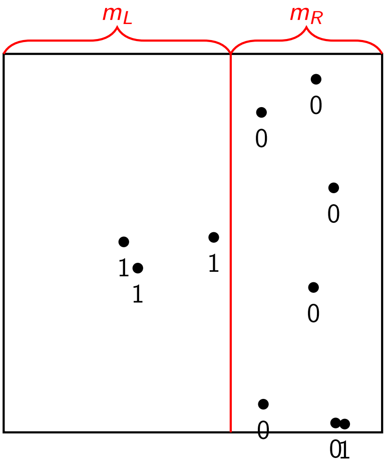
- proportion of observations are

$$p_L = 0.2 \quad \text{and} \quad p_R = 0.8\,.$$

- we deduce

$$\Delta I(m) \approx 0.24 - 0.2\cdot 0 - 0.8\cdot 0.19 = 0.09\,.$$

Quick recap: impurity decrease



- let us look at another candidate
- we compute

$$i(m_L) = 0\,,$$

$$i(m_R) = \frac{1}{2}\left(\frac{6}{7}\cdot\left(1-\frac{6}{7}\right)+\frac{1}{7}\cdot\left(1-\frac{1}{7}\right)\right) \approx 0.12$$

- proportion of observations are

$$p_L = 0.3 \quad \text{and} \quad p_R = 0.7\,.$$

- we deduce

$$\Delta I(m) \approx 0.24 - 0.3\cdot 0 - 0.7\cdot 0.12 = 0.16\,.$$

- this split is much better

Impurity decrease: regression

- slightly different definition in the regression case
- in that case, $Y_i \in \mathbb{R}$ and we look at the (weighted) **variances**
- more precisely:

Definition:[19] For a given node m and split z across feature j, we define

$$L(j,z) := \frac{1}{N(m)} \sum_{i : X_i \in m} (Y_i - \overline{Y}_m)^2 - \frac{1}{N(m)} \sum_{i : X_i \in m} (Y_i - \overline{Y}_{m_L} \mathbb{1}_{X_i^{(j)} < z} - \overline{Y}_{m_R} \mathbb{1}_{X_i^{(j)} \geq z})^2 ,$$

where $\overline{Y}_m$ is the average response on cell m.

- **Intuition:** good split produces cells with constant responses

[19] Scornet, Biau, Vert, *Consistency of random forests*, The Annals of Statistics, 2015

Decision trees: quick recap

- many options for impurity choice / which features are explored
- popular method: **classification and regression trees** (CART[20])
- informally, at a given depth:

```
for $m$ in nodes do
    for $j \in [d]$ do
        for split $\in$ possible splits do
            compute and store $L(j, z)$
        end for
    end for
    split according to $(j^\star, z^\star)$ maximizing $L(j, z)$
end for
```

- **Stopping criterion:** usually `max_depth` / pure leaves

[20] Breiman, Friedman, Olshen, and Stone *Classification and regression trees*, Chapman & Hall, 1984

Mean decrease impurity

- **General idea:** use the numerical criterion to give feature importance

Definition: Let $j \in [d]$. Let $\mathcal{T}$ be a CART tree, $\mathcal{T}_j$ the set of nodes with splits according to feature j. The **mean decrease impurity**[21] is defined as

$$\widehat{\mathrm{MDI}}_j := \sum_{m \in \mathcal{T}_j} p_m \Delta I(m)\,,$$

where p_m is the proportion of data points falling into cell m, and $\Delta I(m)$ is the decrease in impurity at node n.

- **In other words:** $\widehat{\mathrm{MDI}}_j =$ weighed decrease in impurity related to splits along j
- **Intuition:** high if tree uses feature to efficiently split

[21] Breiman, *Random Forests*, Machine Learning, 2001

Nice properties of MDI

- **Empirical variance** of the observations:

$$\widehat{\mathrm{Var}}(Y) := \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y})^2 \,.$$

- for a function $f : [0,1] \to \mathbb{R}$, **train error**

$$R_n(f) := \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(X_i))^2 \,.$$

Proposition:[22] Let $\mathcal{T}$ be a CART tree. Then

$$\widehat{\mathrm{Var}}(Y) = \sum_{j=1}^{d} \widehat{\mathrm{MDI}}_j + R_n(\mathcal{T}) \,.$$

[22] Scornet, *Trees, forests, and impurity-based variable importance*, Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques, 2023

Nice properties of MDI

- **Reminder:** $R^2 =$ percentage of the variance explained by the model
- **Consequence of previous slide:**

$$R^2 = \frac{\sum_{j=1}^{d} \widehat{\mathrm{MDI}}_j}{\widehat{\mathrm{Var}}(Y)} \,.$$

- **Other consequence:** if tree fully-grown, $R_n = 0$ and

$$\widehat{\mathrm{Var}}(Y) = \sum_{j=1}^{d} \widehat{\mathrm{MDI}}_j \,.$$

- **For linear models** ($f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d$) and uniform inputs:

$$\widehat{\mathrm{MDI}}_j \approx \frac{\beta_j^2}{12} \,.$$

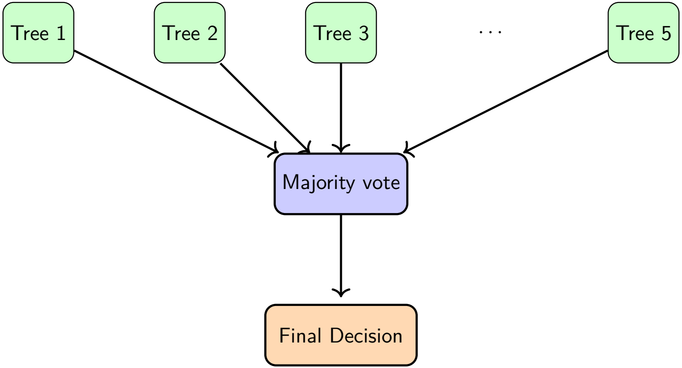
Limitations of MDI

- **Assume:** $Y = f(X) + \varepsilon$, $\mathrm{Var}\left(\varepsilon\right) = \sigma^2$
- then
$$\lim_{n \to +\infty} \sum_{j=1}^{d} \widehat{\mathsf{MDI}}_j = \mathrm{Var}\left(f(X)\right) + \sigma^2 \,.$$
- the sum of MDIs contains not only all available information $\mathrm{Var}\left(f(X)\right)$...
- ...but also noise of the data
- MDI of some variables are higher than expected
- **Other issue:** MDI favors variables with many categories[23]

[23] Strobl et al., *Bias in random forest variable importance measures: illustration, sources and a solution*, BMC Bioinformatics, 2007

Random Forests: quick recap

- **Random forests:**[24] aggregate several trees together
- prediction = mean (regression) or majority vote (classification)



[24] Breiman, ibid.

Random forests: quick recap

- the **random** aspect comes from the construction of each individual tree
- **Tree construction:** for each tree,
  1. sample (with replacement) m points
  2. build a CART tree on these points
- **Additional caveat:** explore only a strict subset of the features at each split
- the points which are not considered in the construction of tree t are called **Out-of-bag (OOB)** points
- typical value: $T = 200$ trees $\to$ not so interpretable anymore
- the user is not going to look at 200 traces
- $+$ potentially conflicting...
- one can still propose simple mechanisms to get interpretability
- let us look into 2 **ad-hoc methods for random forests**

Mean decrease impurity for random forests

- **Idea:** average for all trees in the forest
- **Recall:** for any tree t, we defined

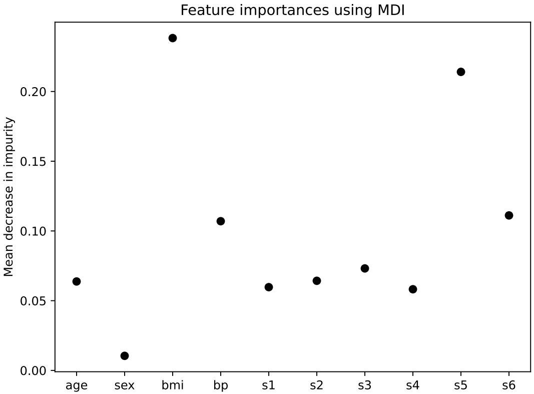
$$\widehat{\mathrm{MDI}}_j(t) := \sum_{m \in t_j} p_m \Delta I(m)\,,$$

  where p_m is the proportion of data points falling into cell m, and $\Delta I(m)$ is the decrease in impurity at node n
- **For random forests:** let $\mathcal{F}$ be a forest

$$\widehat{\mathrm{MDI}}_j(\mathcal{F}) := \frac{1}{T} \sum_{t \in \mathcal{F}} \widehat{\mathrm{MDI}}_j(\mathcal{T})\,.$$

- since taking average, same properties

Mean Decrease Impurity: example



- **Figure:** computing the MDI on the `diabetes` dataset[25]

[25] Efron et al., *Least Angle Regression*, Annals of Statistics, 2004

Summary

- CART trees: iterative splitting according to impurity
- **Mean Decrease Impurity** looks at average decrease for each feature
- gives feature importance of our model
- can be connected to variance of the observations
- can be extended to random forests

5.2. Mean decrease accuracy

Mean decrease accuracy

- **Recall:** in the random forest procedure, each tree is build on a subset of the data
- thrown-away points = out-of-bag (OOB) samples
- **Natural idea:**[26] take advantage of these points
- **Mean decrease accuracy**, a.k.a. permutation-based feature importance
- **More precisely:** for each tree t, for each feature j,
  1. permute values of column j for the OOB samples
  2. compute prediction of tree t for these new points
- we then compare the predictions with the ground-truth
- report the increase in misclassification per feature
- **Intuition:** if j important in every tree, permuting the values *breaks* the predictor

[26] Breiman, *Random Forests*, Machine Learning, 2001

MDA: formal definition

- we can be more formal:

Definition (Breiman-Cutler MDA):[27] Let $X_{i,\pi_{j,t}}$ be the ith permuted OOB sample for tree t. We define

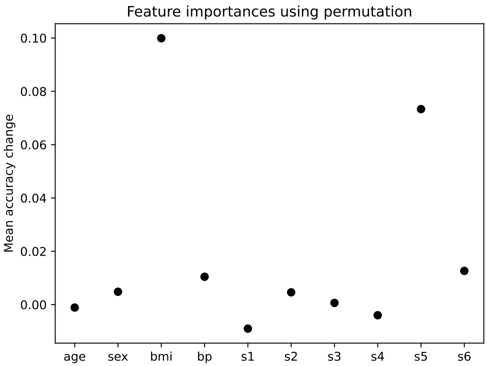
$$\widehat{\text{MDA}}_j := \frac{1}{T} \sum_{t \in \mathcal{F}} \frac{1}{N(t)} \sum_{i \in \text{OOB}(t)} \left[(Y_i - t(X_{i,\pi_{j,t}}))^2 - (Y_i - t(X_i))^2 \right] ,$$

where $N(t)$ is the size of the OOB sample for tree t.

- **Remark:** other definitions are possible

[27] Bénard et al., *Mean decrease accuracy for random forests: inconsistency, and a practical solution via the Sobol-MDA*, Biometrika, 2022

Permutation-based feature importance: example



- **Figure:** computing permutation-based importance on the `diabetes` dataset

Properties of MDA

- assume $Y = f(X) + \varepsilon$
- **For large n:**

$$\widehat{\mathsf{MDA}}_j \longrightarrow \mathrm{Var}\,(Y) \times \mathsf{ST}_j + \mathrm{Var}\,(Y) \times \mathsf{ST}_j^{\mathsf{mg}} + \mathsf{rest}\,,$$

  where ST is the Sobol total index[28]
- **Sobol index** ≈ contribution to the output variance of the main effect feature j
- **Problem:** “rest” can be large and does not correspond to anything meaningful...

[28] Sobol, *Sensitivity estimates for nonlinear mathematical models*, Math. Mod. Comp. Exp., 1993

Summary

- for some models, we can **take advantage of the internal mechanics**
- still no obvious choice (many possibilities!)
- in the case of **random forests**, we have seen two possibilities:
  - **Mean Decrease Impurity** averages decrease in impurity for nodes containing the feature
  - **Permutation-based feature importance** permutes inspected feature values and looks at drop in accuracy