DataScientist Society

# 決定木によるクラス分類

## ～決定木モデル構築, 可視化～

# 今回の目標

- 決定木の概要理解
- 決定木の作成，可視化を繰り返して精度を向上させる
- 予測結果を投稿して，未知データでの精度を確かめる

# クラス分類と問題設定

## ➢ 今回扱う問題

✓ 顧客が定額預金の申し込みをするかを予測
→ 顧客を「申し込む」「申し込まない」の2つのクラスに分類

✓ 予測モデル
→ 入力データをもとに，「申し込む」クラスに属する確率を出力

## ➢ 出力値の使い方

✓ 重病を予測するモデルでは重病確率が10%もあればアラートを出して精密検査を推奨するべき

✓ 株価の上昇下落を予測するモデルでは上昇確率が50%を超えていないと買い注文を出せない

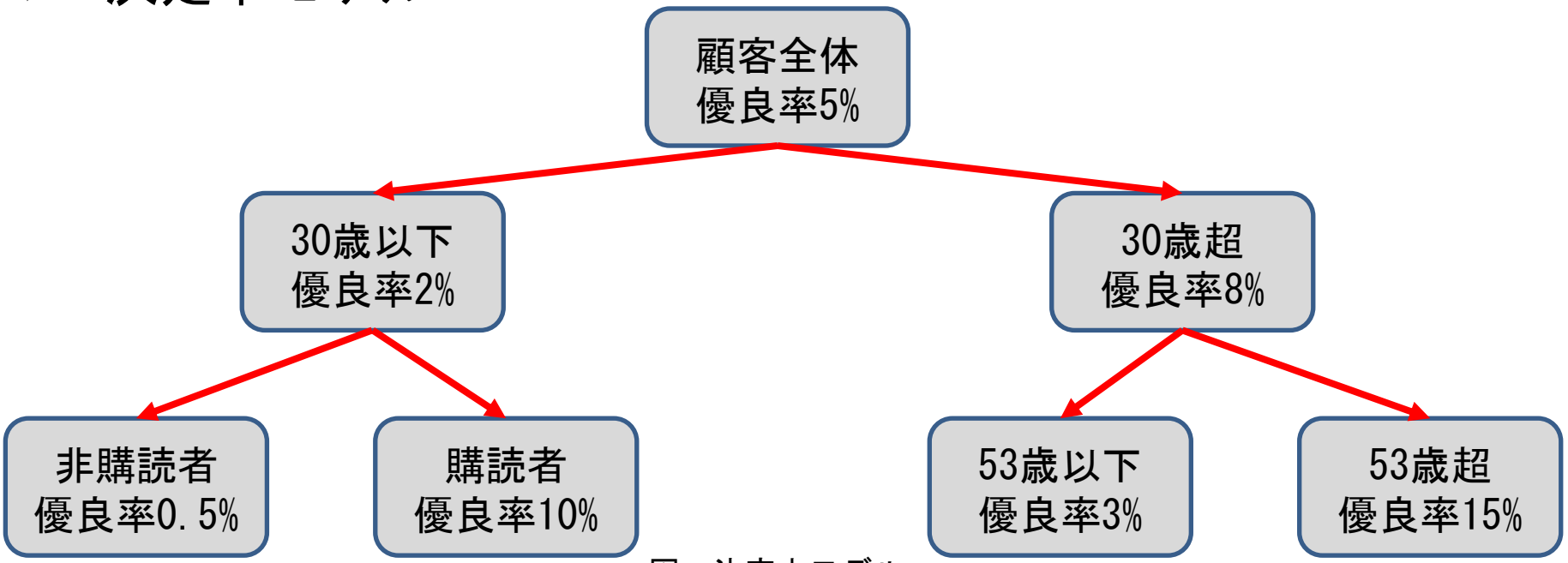
※出力値の扱いは問題設定に依存する，講座では確率を算出するとこまでを扱う.

# 構築したいモデル

- 問題設定

  あるお店の顧客データとして「優良顧客フラグ」「年齢」「メールマガジン購読フラグ」「前回来店日」「前々回来店日」のデータがあり「優良顧客」を予測するモデルを決定木で構築する.

- 決定木モデル




図：決定木モデル

# 決定木モデルの構築概要

## ➢ 分割基準の選び方

- ✓ データを全説明変数に対して閾値を虱潰しに探索し，スコア（※後述）の最も良いもので分割する
- ✓ これを分割後のデータに対しても繰り返し行い木を作成する

表：スコア探索

| 説明変数 | 閾値 | 左側優良率 | 右側優良率 | スコア |
|---|---|---|---|---|
| 年齢 | 10 | 0.10 | 0.07 | 0.05 |
| 年齢 | 11 | 0.05 | 0.15 | 0.06 |
| … | … | … | … | … |
| 年齢 | 30 | 0.01 | 0.20 | 0.10 |
| … | … | … | … | … |
| 年齢 | 65 | | | |
| 購読フラグ | 0 | 0.10 | 0.20 | 0.05 |
| 前回来店日 | 6月1日 | 0.05 | 0.06 | 0.01 |
| 前回来店日 | 6月2日 | 0.06 | 0.05 | 0.01 |
| … | … | … | … | … |
| 前回来店日 | 9月30日 | 0.05 | 0.06 | 0.01 |
| 前々回来店日 | 5月1日 | 0.06 | 0.05 | 0.01 |
| 前々回来店日 | 5月2日 | 0.05 | 0.06 | 0.01 |
| … | … | … | … | … |
| 前々回来店日 | 9月29日 | 0.05 | 0.06 | 0.01 |

※数字は適当です

# 決定木モデルの構築概要

## ➢ 分割スコアの計算方法

分割スコアとしてよく使われるのは，下記の2つである．このとき，$p_S$をセグメント$S$の一方のクラス所属確率として，

- ✓ Gini

$$I_G(S) := 1 - p_S^2 - (1 - p_s)^2$$

- ✓ Entropy

$$I_E(S) := -p_S \log(p_S) - (1 - p_S)\log(1 - p_S)$$

Gini，Entropyの両方とも，$p_S$が0または1となるときに，最小値となる．いま，分割前のデータを$S_0$，分割後のデータを$S_L, S_R$，分割後に$S_L, S_R$に所属する確率を$p_L$，$p_R$として，

$$I_G(S_0) - p_L I_G(S_L) - p_R I_G(S_R)$$

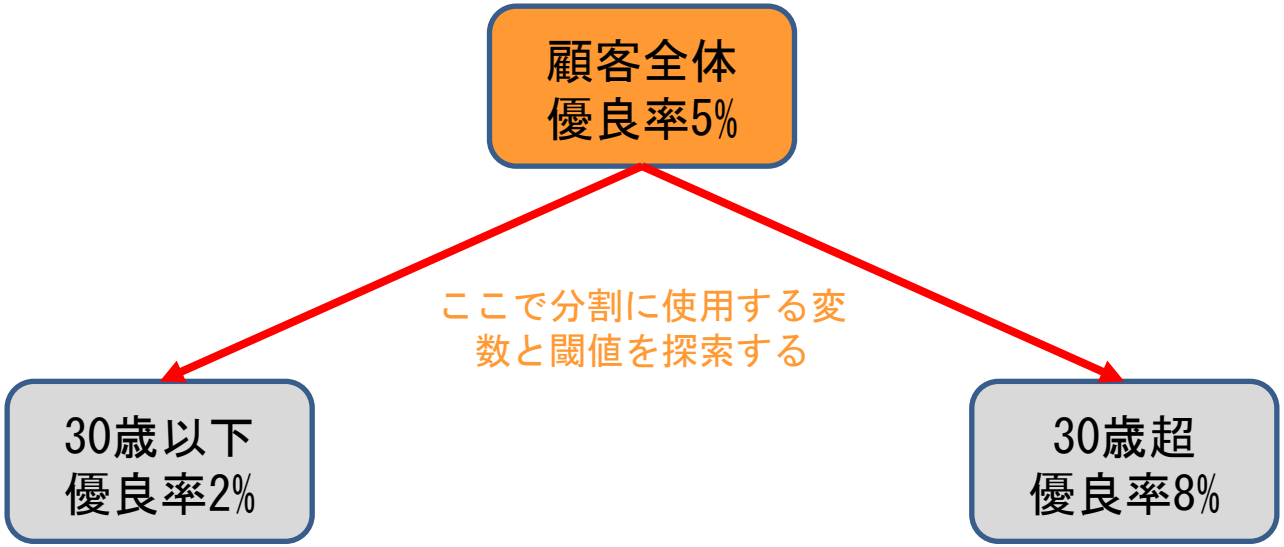
or

$$I_E(S_0) - p_L I_E(S_L) - p_R I_E(S_R)$$

が最大となるような説明変数と閾値を分割基準とする．

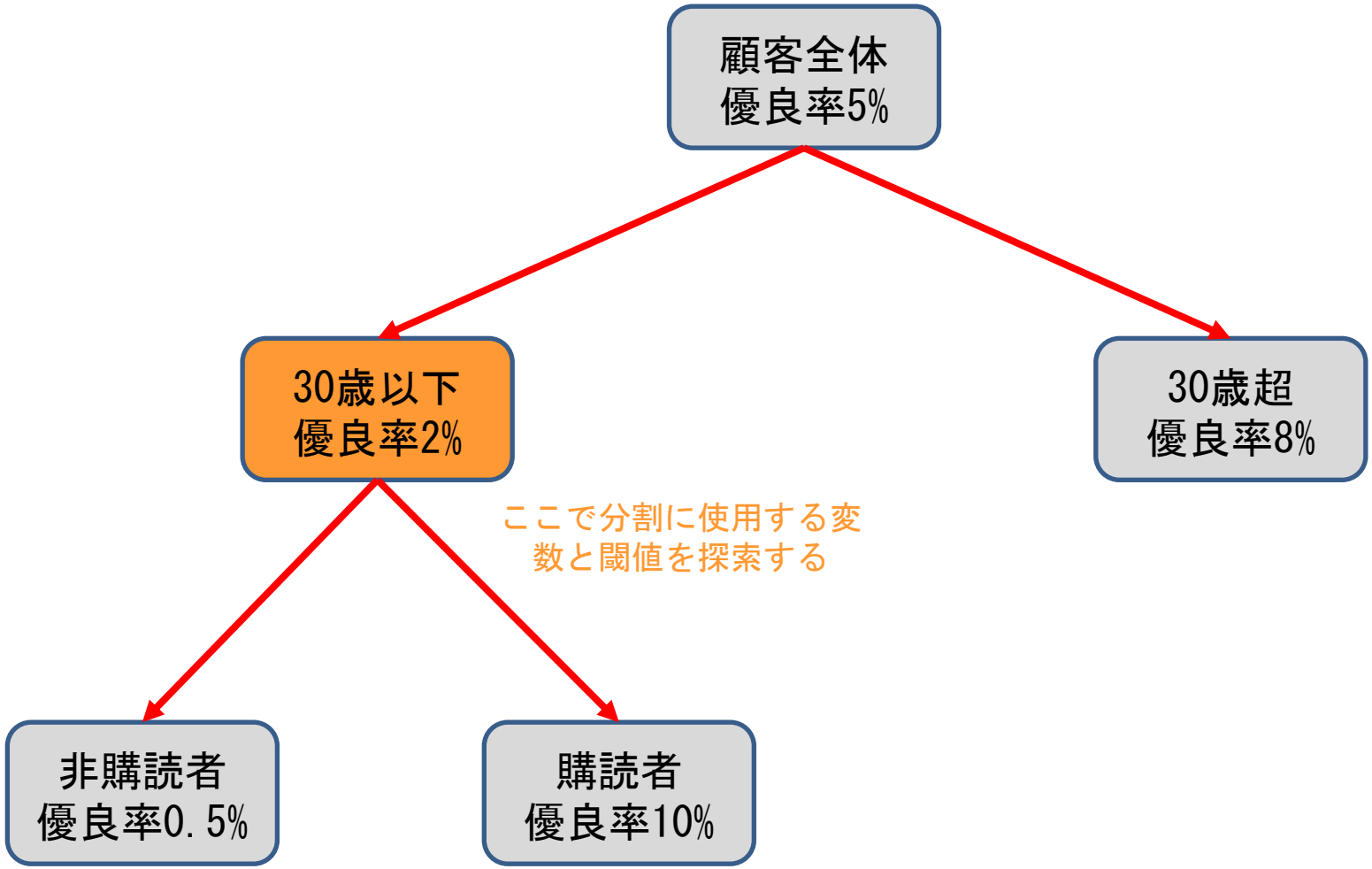
# 決定木モデルの構築概要

- 構築過程の理解



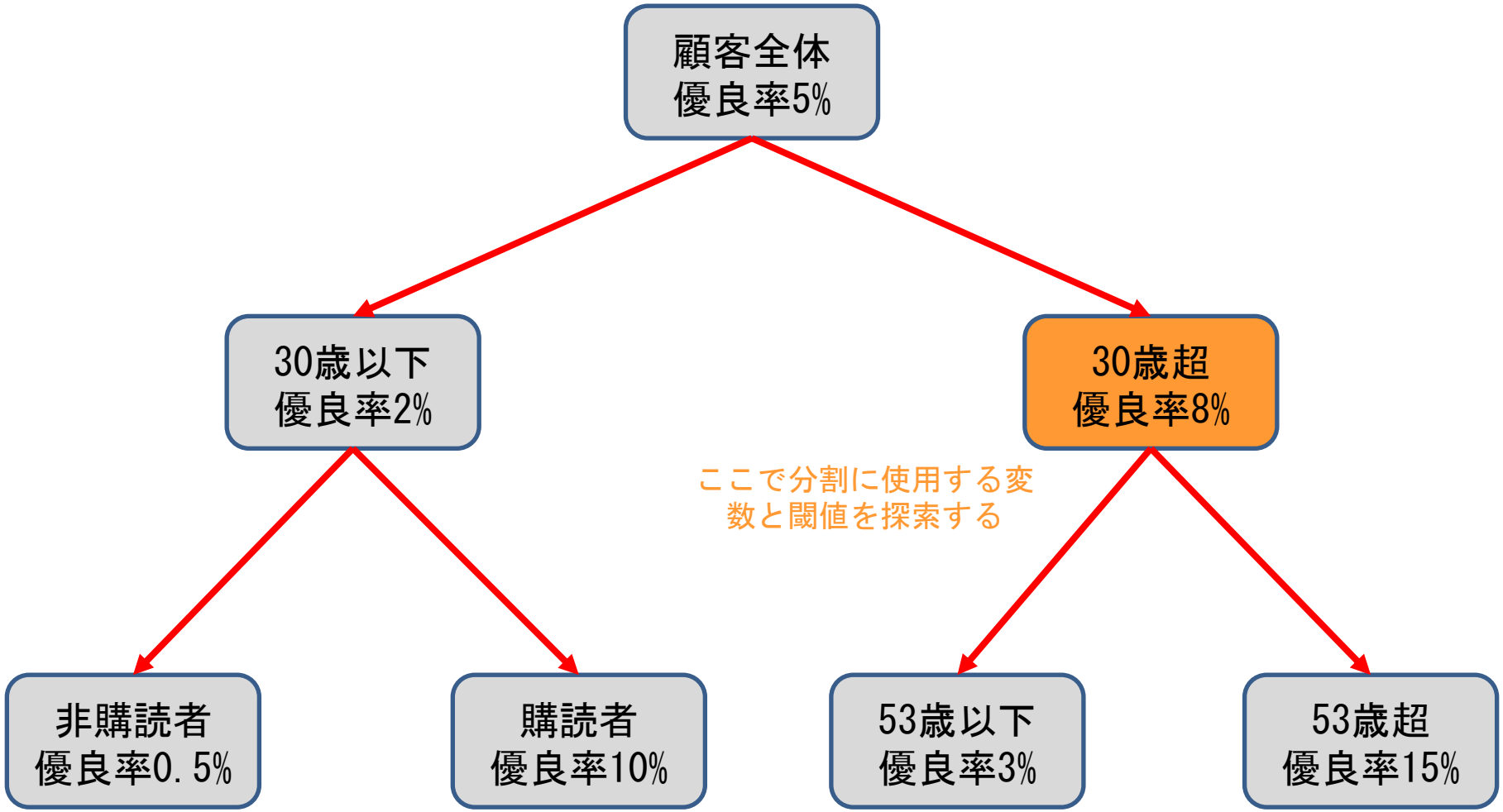
# 決定木モデルの構築概要

- 構築過程の理解



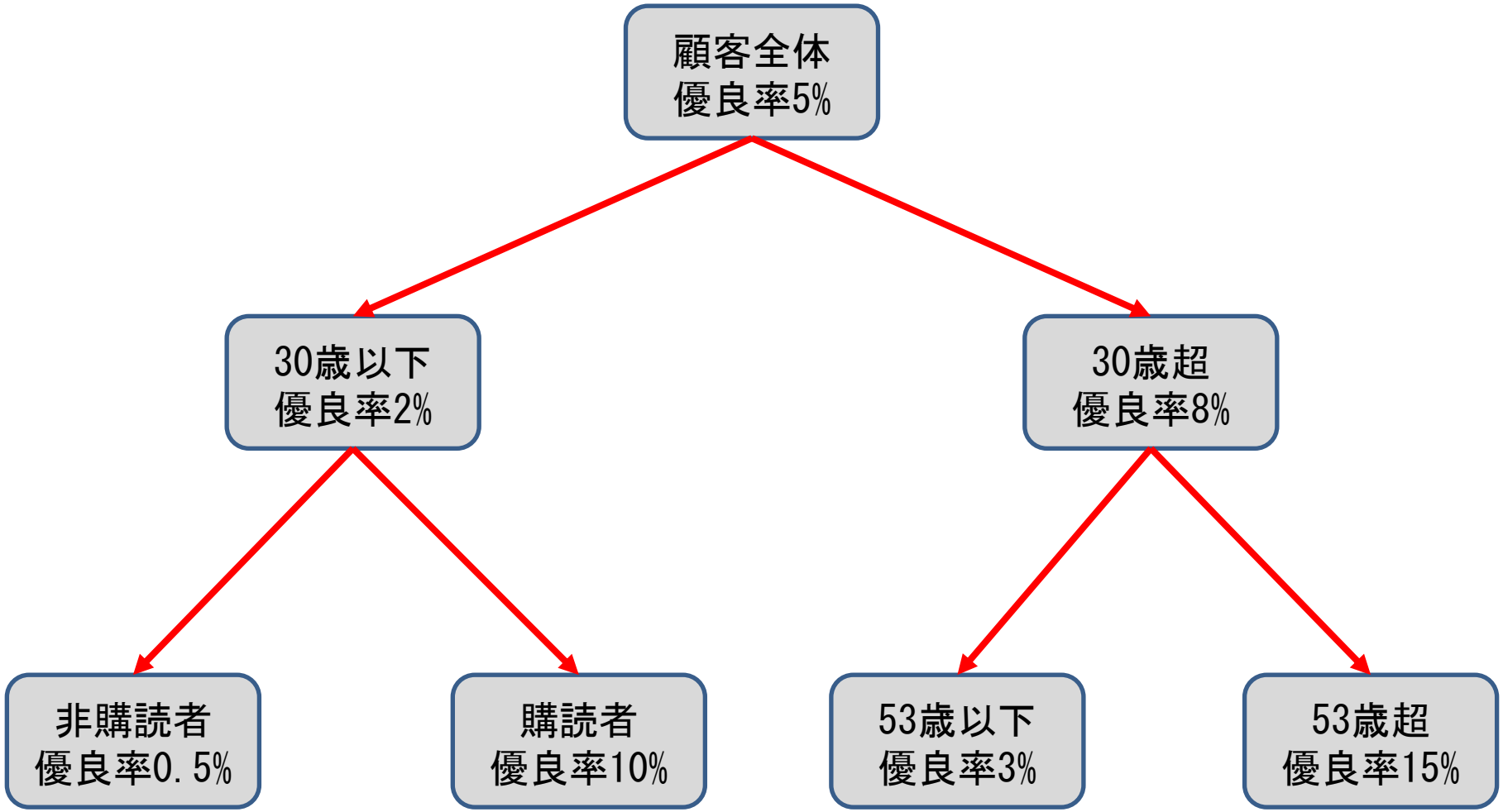
# 決定木モデルの構築概要

- 構築過程の理解



# 決定木モデルの構築概要

- 構築過程の理解



# 決定木モデルの特徴

- 分かりやすい
  - ✓ 図で表現できるので，担当者や顧客へ説明がしやすい
- 交互作用を捉えることができる
  - ✓ あくまで単変数ごとに探索を行っているが，副次的に交互作用が現れることがある
- 複数変数の線形結合には弱い
  - ✓ 前回来店日と前々回来店日の差を学習するには，

    来店間隔 := 前回来店日 - 前々回来店日

    を新しい説明変数として追加する必要がある．

# 決定木モデルの学習パラメータ(Python/R)

決定木の中で主要なものを説明する.

- ✓ 木の深さ(max_depth/maxdepth)
  値を大きくすると細かく分割して，表現能力が高くなる．分割数は最大で$2^{\text{max_depth}}$となるため，通常は10程度に抑えることが多い.

- ✓ 最小ノードサイズ(min_samples_leaf/minbucket)
  セグメントに含まれるデータ数の下限値．大き目の値を設定することで，特異な例での分割を避けることができる.

- ✓ 枝刈り(剪定)の強さ(ccp_alpha/cp)※scikit-learnはver0.22より実装(2019年12月～)
  決定木は分割後にセグメント間で少しでも差があると分割を実行するが，有意でない分割を行わないための閾値．大き目の値を設定することで，特異な例での分割を避けることができる.

パラメータを変えることで，どのように構築される決定木が変わるのかを演習で確認してください.

# 決定木モデルの学習パラメータ(Python/R)

枝刈り(剪定)の強さは「木の深さ」や「最小ノードサイズ」などと違い，何の値なのかが不明瞭なので，ここで概要の補足説明をする．
※公式ドキュメントより引用：詳細が気になる人はここを参照してください
https://scikit-learn.org/stable/modules/tree.html#minimal-cost-complexity-pruning

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|$$

作成した決定木の
「末端ノードのスコア(不純度)の合計」

作成した決定木の
「末端ノード数」

この$R_\alpha(T)$を最小にするように末端ノード数を減らしていく．このとき，スコア(不純度)は末端の方が小さくなるように作られていたので，分岐(枝)を減らして末端ノード数を減らすと，$R(T)$は大きくなるので$\alpha(\geq 0)$の値によって減らし具合が変わってくる(デフォルトは$\alpha = 0$で枝刈りはされない)．

# 引用, 参考文献

[1] 平井有三：はじめてのパターン認識, 2012年, 森北出版株式会社