予測問題発展: モデル集計

機械学習入門

川田恵介

Table of contents

1 復習

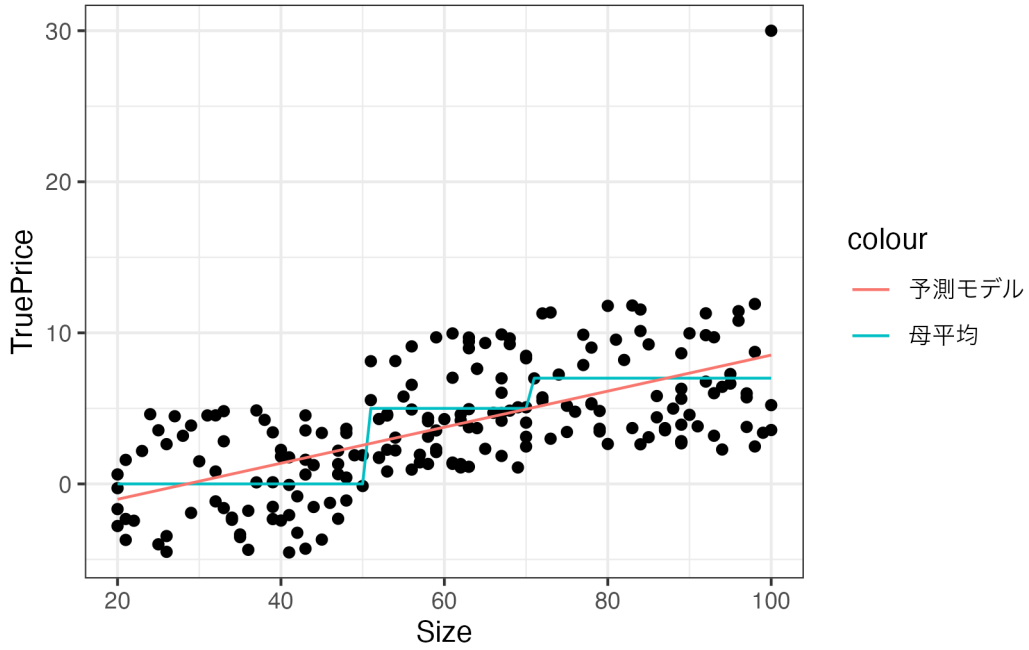
1.1 予測問題

- X から Y を予測する
- 「$(Y -$ 予測モデル$)^2$ の平均値」を予測性能の指標とするのであれば、
  - Y の母平均が理想的な予想モデル

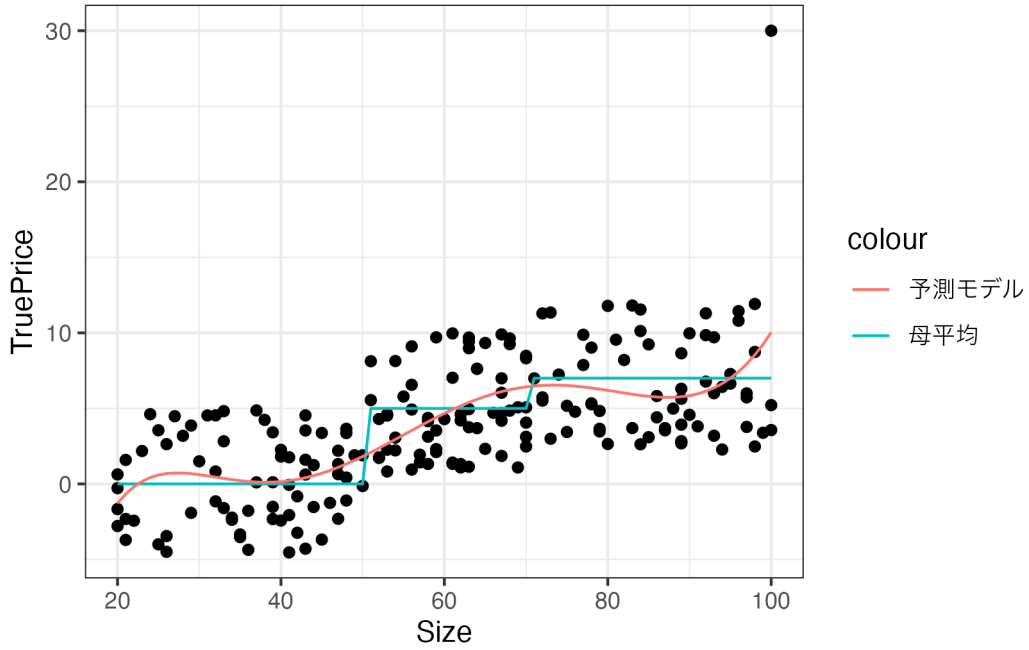
1.2 OLS と LASSO

- 予測モデルの大枠 $\beta_0 + \beta_1 X_1 + .. + \beta_L X_L$ を**研究者**が設定、β をデータにより決定
- OLS: 複雑にしすぎる (β の数を増やしすぎる) と、
  - データに過剰適合
  - 母平均から乖離
  - 予測性能が悪化
- LASSO: **データ主導**で、複雑さを抑制する

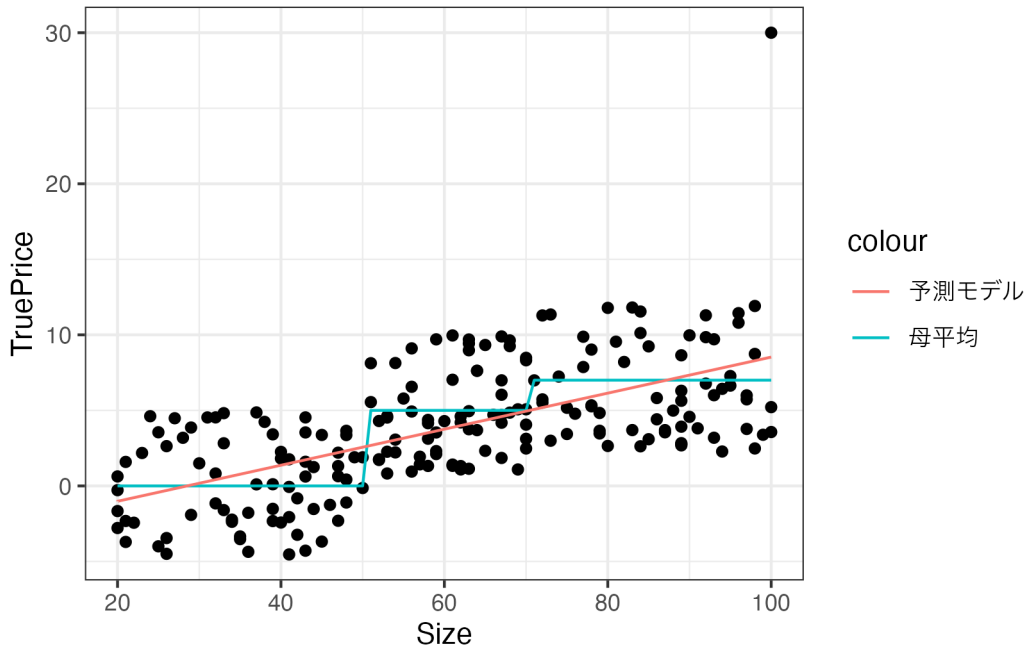
1.3 例: OLS: Price ~ Size



1.4 例: OLS: Price ~ Size + .. + Size^6



1.5 例: LASSO: Price ~ Size + .. + Size^6

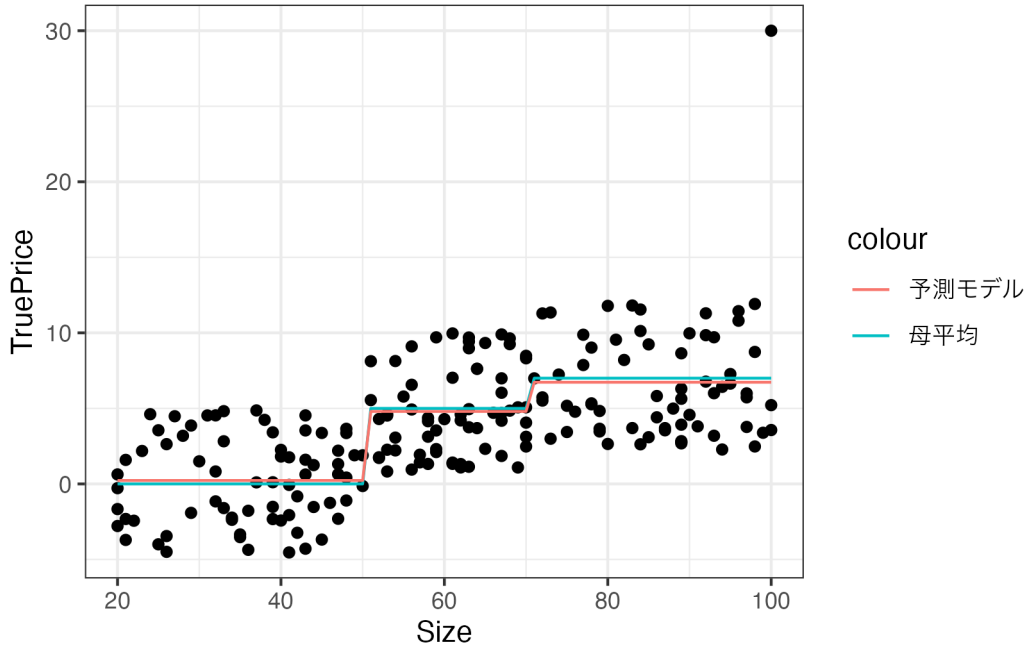


1.6 決定木 モデル

- OLS や LASSO は線型モデル ($\beta \times X$ の足し算のモデル) と呼ばれる分類
  - 母平均によっては、異なってモデルの方が適する
  - 代表例は決定木 (サブグループ平均) モデル (01Introduction 参照)
    - サブグループ平均値を予測値とする

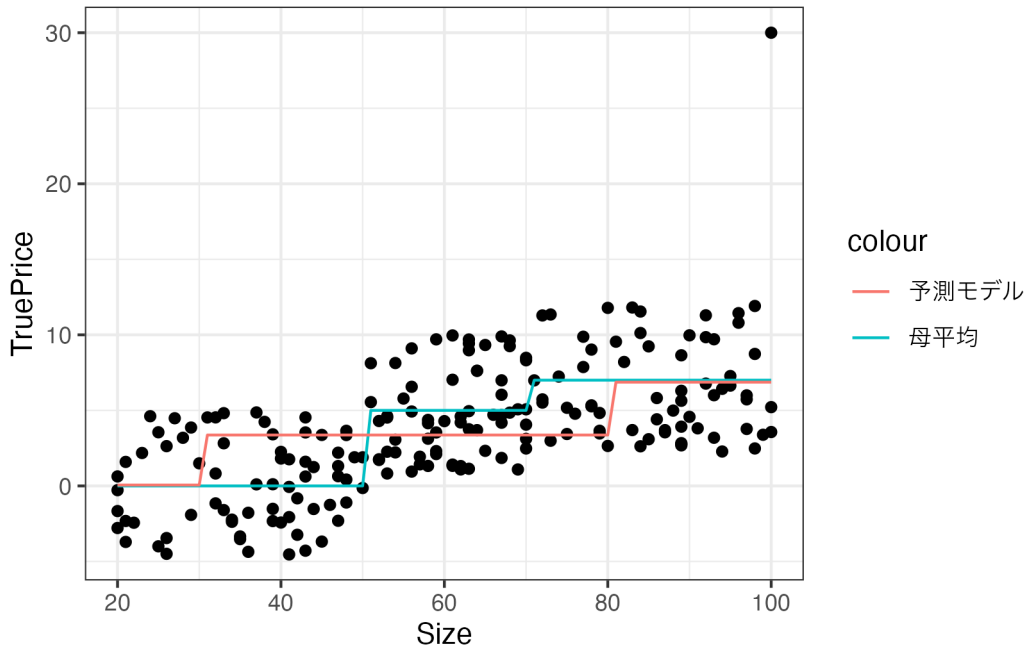
1.7 例: 決定木

- Size $\in [-50, 50-70, 70-]$ に分割し、平均値を計算



1.8 例: 決定木

- Size $\in [-30, 30-80, 80-]$ に分割し、平均値を計算

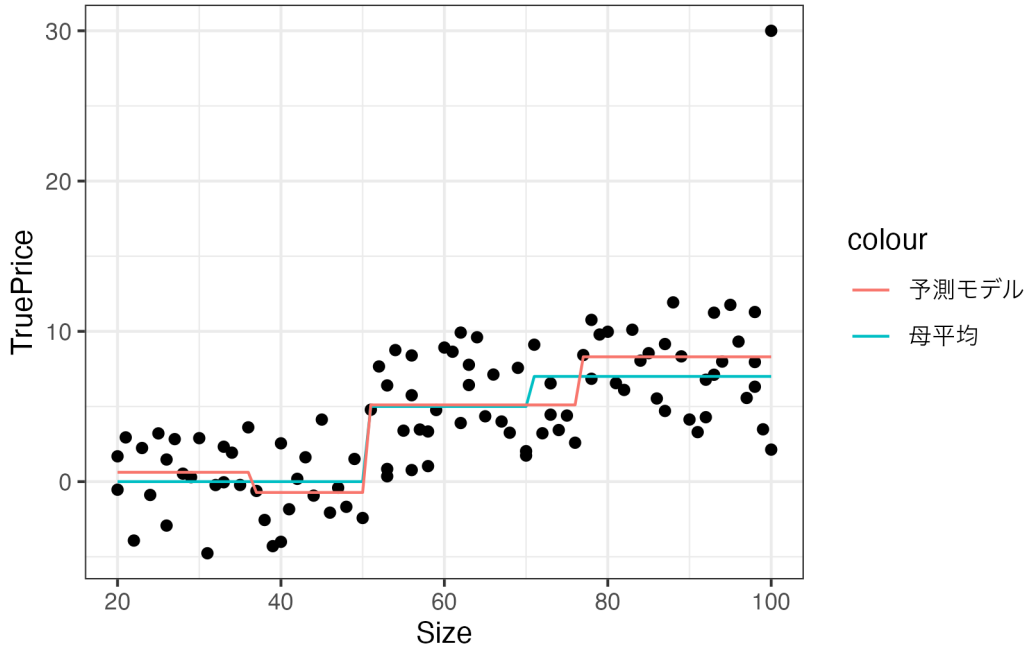


1.9 決定木の難しさ

- サブグループの定義に予測性能が決定的に依存する
  - OLS と類似
- データ主導の決定木
  - データへの適合度を最大化するようにグループの定義を決定

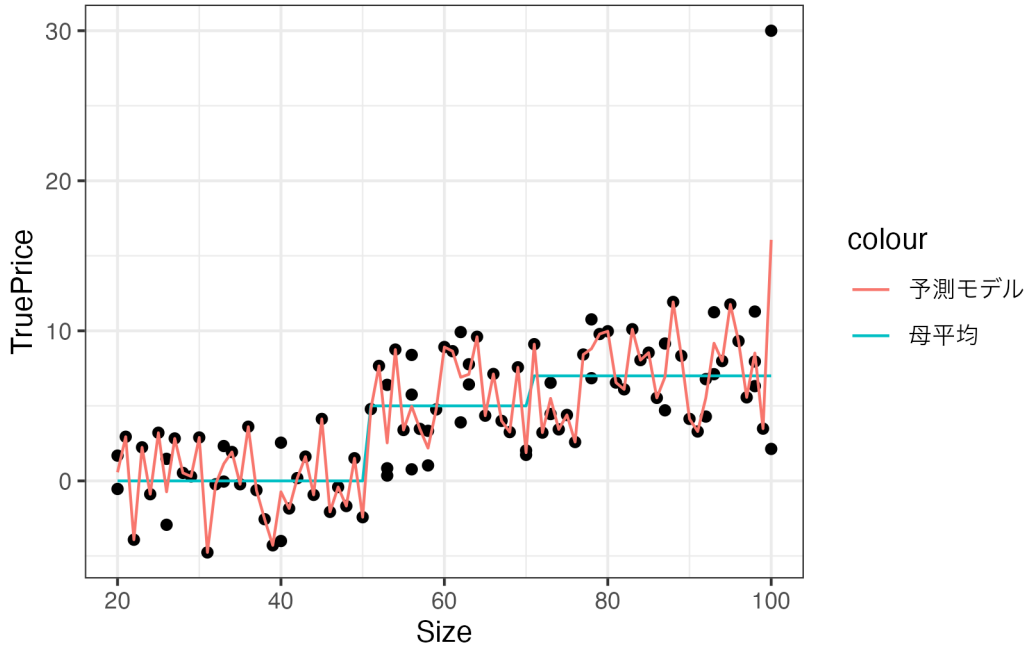
1.10 例: データ主導の決定木

- 最大 2 分割、最小サンプルサイズ = 5



1.11 例: データ主導の決定木

- 最大 30 分割/最小事例数 1



1.12 複雑さの影響

- 複雑なモデル = 分割数が多い
  - 少数事例の平均値を予測値となるので、事例の偏りの影響が大きい
    - 母平均から乖離する可能性が高くなる
- LASSO とは異なり、モデルの単純化は行わない
  - 何らかの方法で単純化する必要がある
    - 剪定などの多様な方法があるが、ここではモデル集計を紹介

2 モデル集計: Random Forest

- 複雑なモデルを推定すると、母平均から大きく乖離した事例 (ハズレ値) の影響を強く受ける
  - 多様な予測モデルを沢山作り、その予測の平均値を最終予測とする

2.1 直感

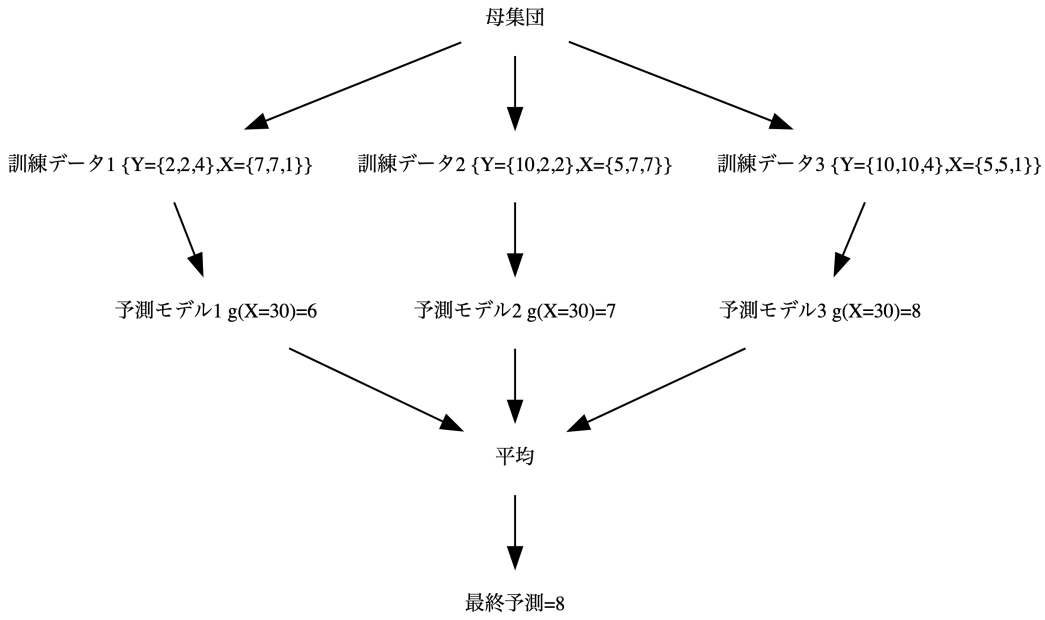
- 「不適当な予測」が生じる大きな原因は、「観察した事例の偏り」
  - どれだけ「情報処理」に長けた「専門家」でも、偏った経験により、極端な予測を行いうる

  - 誰にこのような現象が生じているのか、事前にはわからない

- 解決策: 複数の専門家の予測を集計し、極端な予測の影響を緩和する

2.2 イメージ: モデル集計

- 複数データを用いて、予測モデルを作る

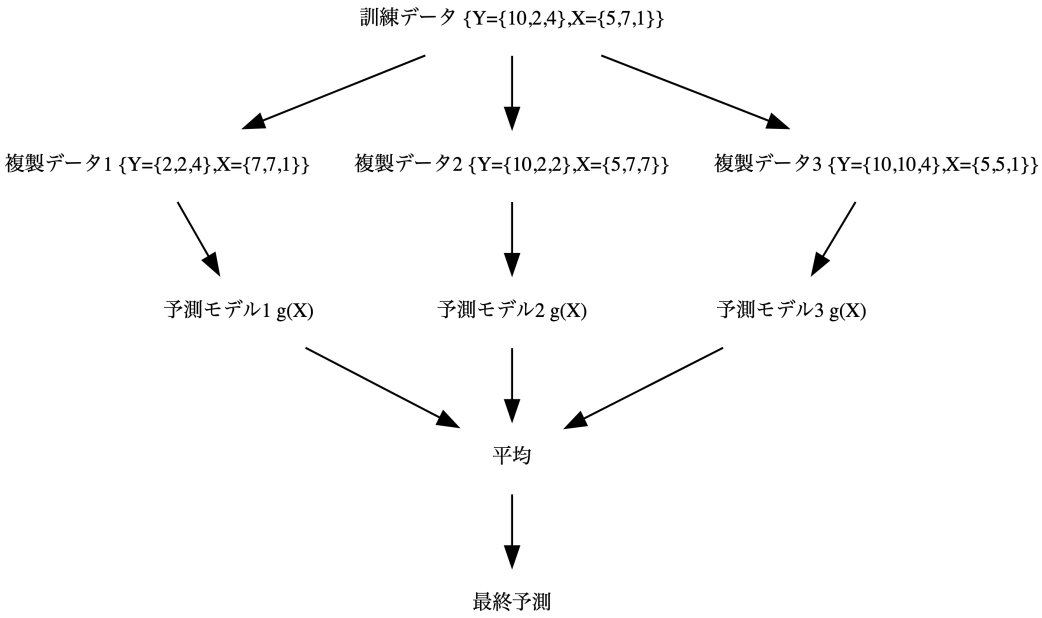


2.3 ブートストラップ集約法 (Bagging)

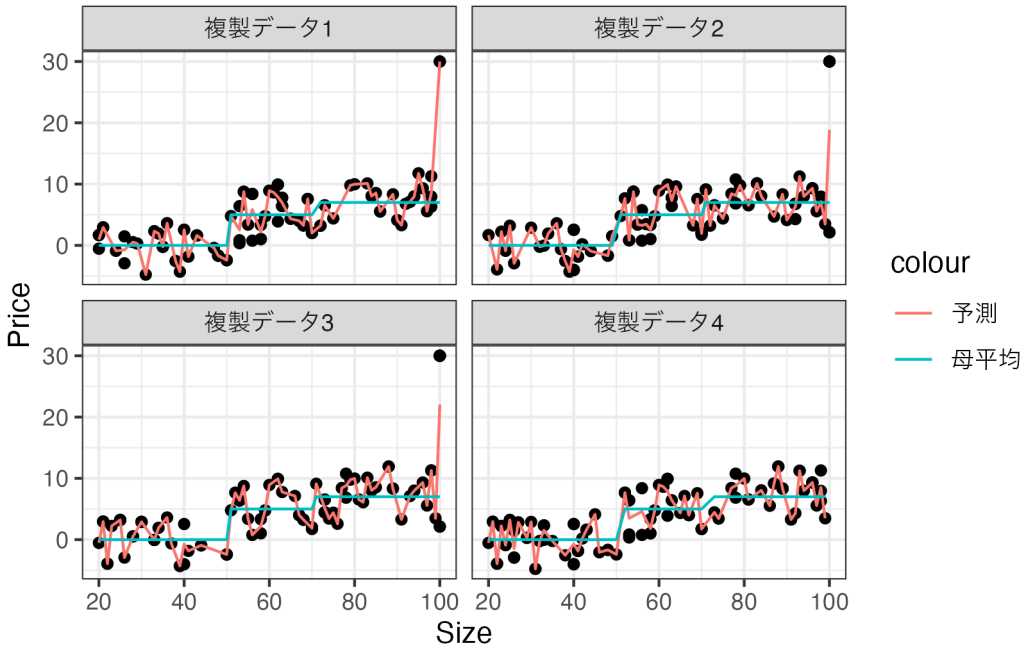
- 現実には、データは一つしか存在しない
  - 訓練データからデータの複製を行い (ブートストラップ)、複製データから生成した予測モデルの平均値を算出する
- ブートストラップ: あるデータから、ランダムに事例を抜き出し、新しいデータを作る
  - 同じ事例を抜き出すことを許容する (復元抽出)

2.4 イメージ: Bagging

- Bootstrap 法を用いて、予測モデルを大量に作る (500-5000 個程度)


訓練データ {Y={10,2,4},X={5,7,1}}
複製データ1 {Y={2,2,4},X={7,7,1}}
複製データ2 {Y={10,2,2},X={5,7,7}}
複製データ3 {Y={10,10,4},X={5,5,1}}
予測モデル1 g(X)
予測モデル2 g(X)
予測モデル3 g(X)
平均
最終予測

2.5 例: Bagging


複製データ1
複製データ2
複製データ3
複製データ4
Price
Size
colour
予測
母平均

2.6 多様な予測の集計

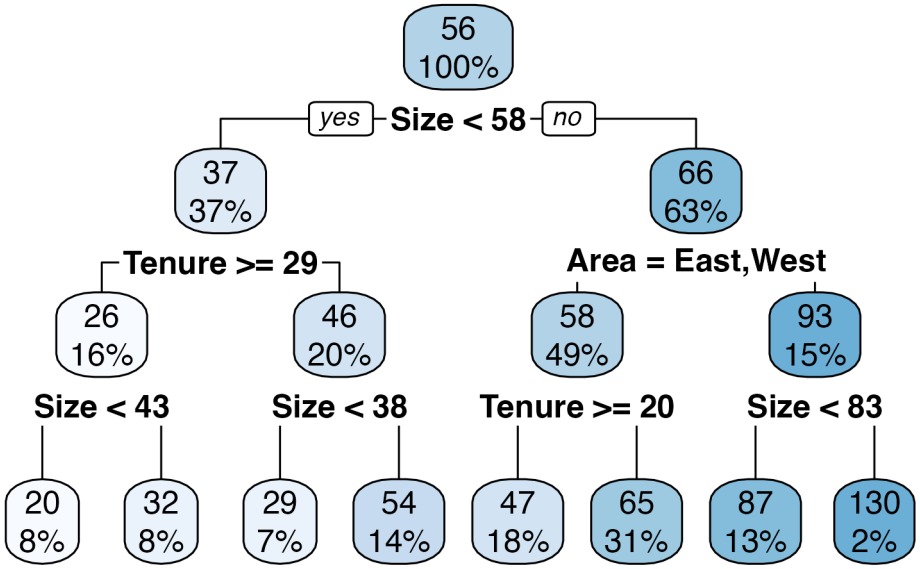
- 一般に、ある程度の予測性能を前提に、多様な予測モデル (データの異なる情報を活用したモデル) を集計した方が性能改善が期待できる
  - 全く同じ予測しか生み出さないモデルを集計しても意味がない

2.7 Bagging の問題

- 多くの応用で、一部の変数が強力な予測力を持つ
  - 不動産価格予測では Size
- 限られた事例数のもとでは、他の (予測力をもつ) 変数の情報が活用されない

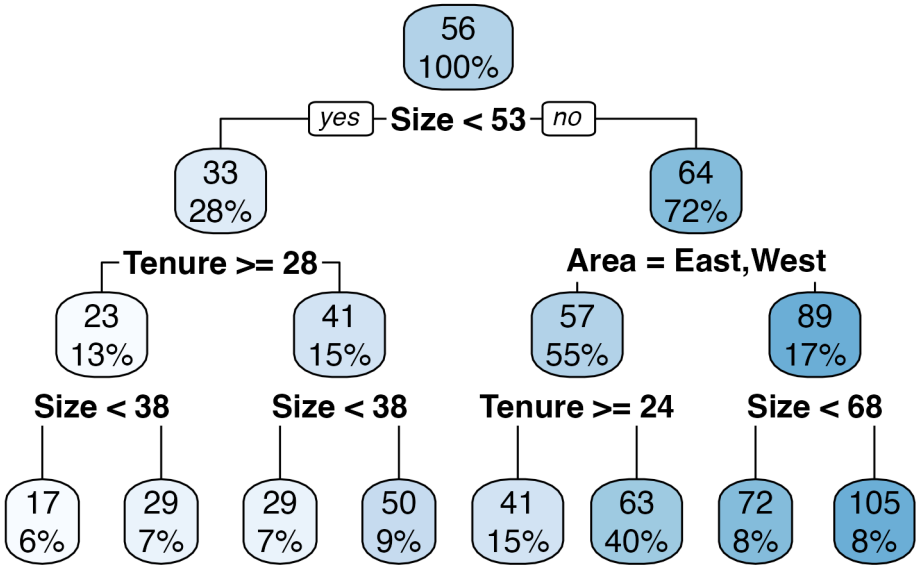
2.8 イメージ: Bagging

- 複製データ 1



2.9 イメージ: Bagging

- 複製データ 2

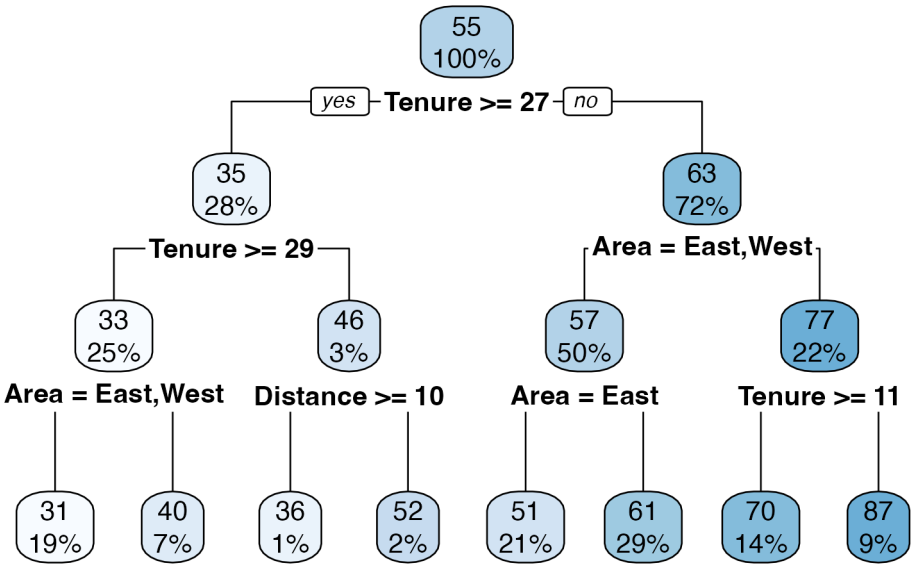


2.10 RandomForest

- 未活用の情報を利用するために、分割に使用できる変数もランダムに決める
  - 典型的には $\sqrt{\text{元の変数数}}$ 個の変数をランダムに選び、その中から分割に用いる変数を探す

2.11 イメージ: RandomForest

- Size の使用を禁止すると



2.12 RandomForest の利点

- Bootstrap は、“ハズレ値” を含まないデータとそれを用いたモデルも生成される
  - ハズレ値の影響を軽減できる
- 変数の一部を確率的に使用できなくすることで、モデル間で用いる変数の多様性が促進される
  - より多くの変数の情報が活用できる
- 注: モデル単位ではなく、サンプル分割ごとに、(元々の変数数の square root 個) ランダムに使用禁止している

2.13 実装

- ranger 関数 (ranger パッケージ) を使えば、lm と同じような文法で実装できる

```
library(ranger)

Group = sample(
  1:2,
  nrow(Data),
  replace = TRUE)

Train = Data[Group == 1,]
```

```
Test = Data[Group == 2,]

RF = ranger(Price ~ Size + Tenure + Distance, Train) # モデル推定

predict(RF, Test)$predictions # 予測値の算出
```

3 モデル集計: Stacking

- 「多様な予測モデルの集計」をさらに推し進める
- 異なるアルゴリズムが生み出すモデルを集計する

3.1 動機

- 予測モデルを生み出す多くのアルゴリズム (OLS/LASSO/Random Forest 等) が提案されている
  - モデル選択: どのアルゴリズムが最も適しているのか?
  - モデル集計: どのように予測を組み合わせれば良いのか?

3.2 復習: モデル選択

- データを 2 分割 (訓練/テスト) し、訓練モデルで予測モデル群を推定し、テストデータで評価する
- 最も性能の良いモデルを選択し、使用する

3.3 モデル集計

- テストデータへの当てはまりが最も良くなるように、モデルを組み合わせる
- 代表的な方法は、線型モデルの推定

$$\beta_{OLS} \times OLSの予測値 + \beta_{RF} RandomForestの予測値 + ...$$

- β をテストデータへの当てはまりを最大化するように推定

3.4 実例

```
library(tidyverse)
library(ranger)
library(hdm)
```

```
Data = read_csv("Public/Example.csv")

Group = sample(
  1:2,
  nrow(Data),
  replace = TRUE
)

Train = Data[Group == 1,]
Test = Data[Group == 2,]
```

3.5 予測

```
OLS = lm(Price ~ Size + Distance + Tenure, Train)
Test$PredOLS = predict(OLS, Test)

LASSO = rlasso(Price ~ (Size + Distance + Tenure)**2 +
                 I(Size^2) + I(Distance^2) + I(Tenure^2),
               Train)
Test$PredLASSO = predict(LASSO, Test)[,1]

RF = ranger(Price ~ Size + Distance + Tenure, Train)
Test$PredRF = predict(RF,Test)$predictions
```

3.6 Stacking

```
lm(Price ~ PredOLS + PredLASSO + PredRF, Test)
```

```
Call:
lm(formula = Price ~ PredOLS + PredLASSO + PredRF, data = Test)

Coefficients:
(Intercept)       PredOLS     PredLASSO        PredRF
    -1.8714       -0.3600        0.7133        0.6736
```

- Random Forest の結果を強く反映したモデル
  - OLS や LASSO の結果も活用

3.7 他の実装

- Stacking の実装については、多くの議論が積み上げられている
  - 交差推定の活用 (後述) など