

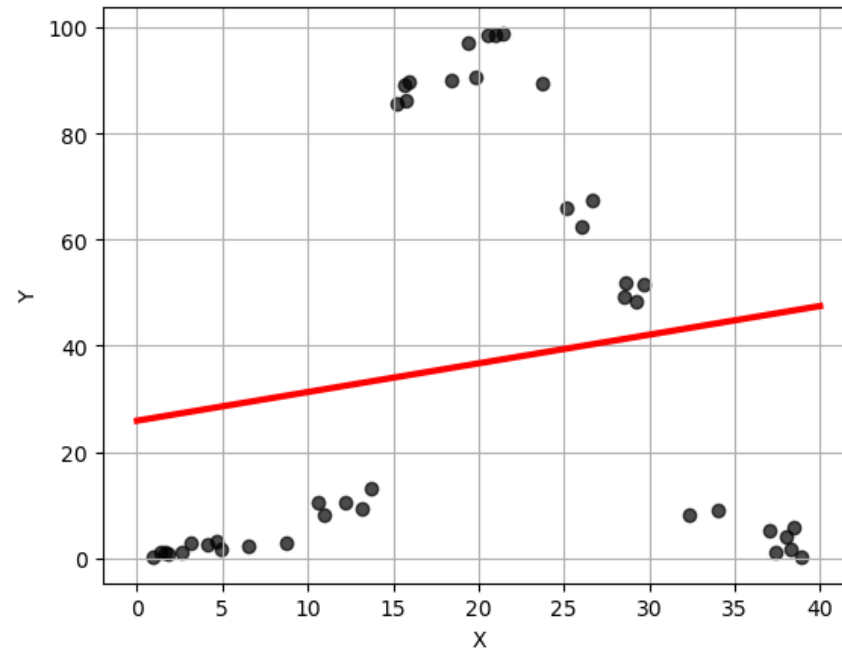
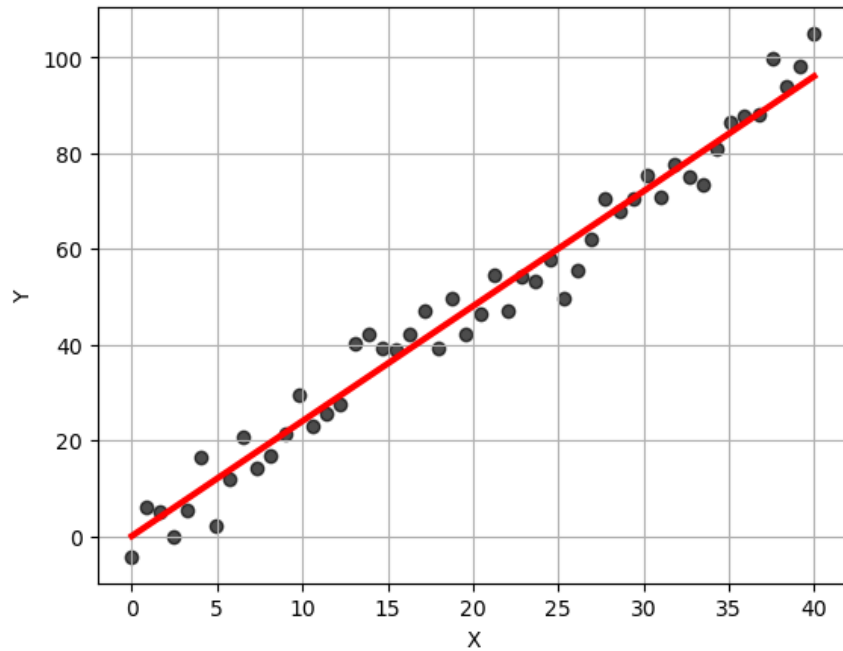
دوره
علم داده،
یادگیری ماشین و
هوش مصنوعی

جلسه سیزدهم:
الگوریتم درخت تصمیم،
جنگل تصادفی و تقویت
گرادیان

ارائه‌کننده:
دکتر فرزاد مینویی

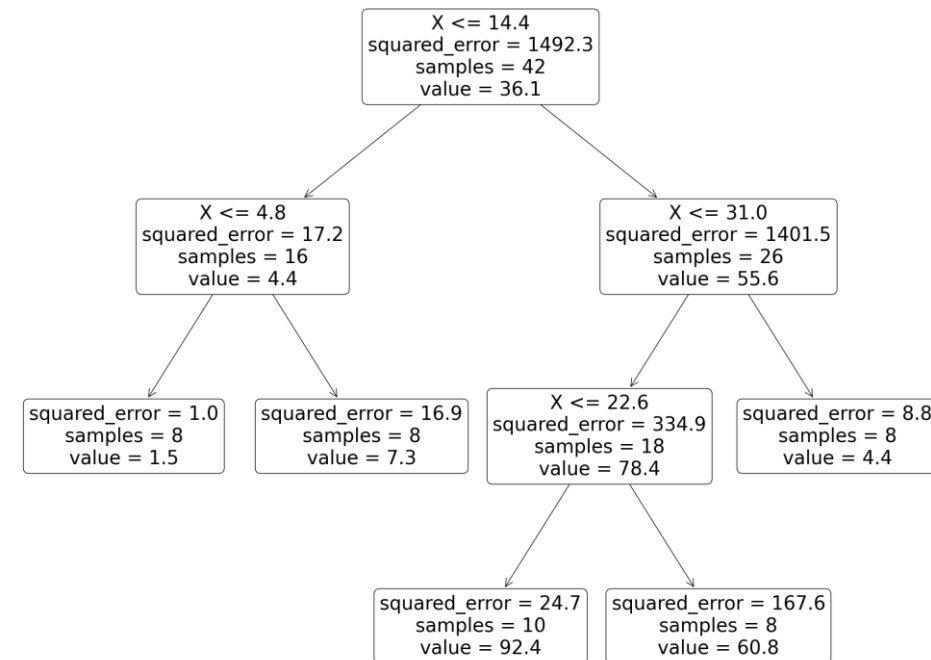
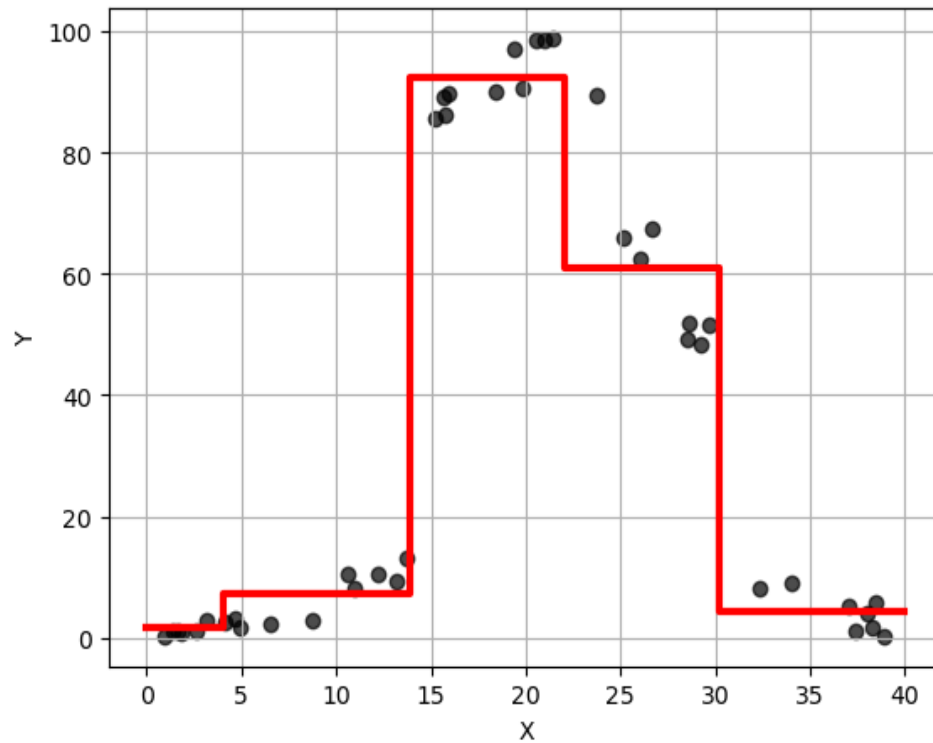


مدل سازی روابط غیرخطی

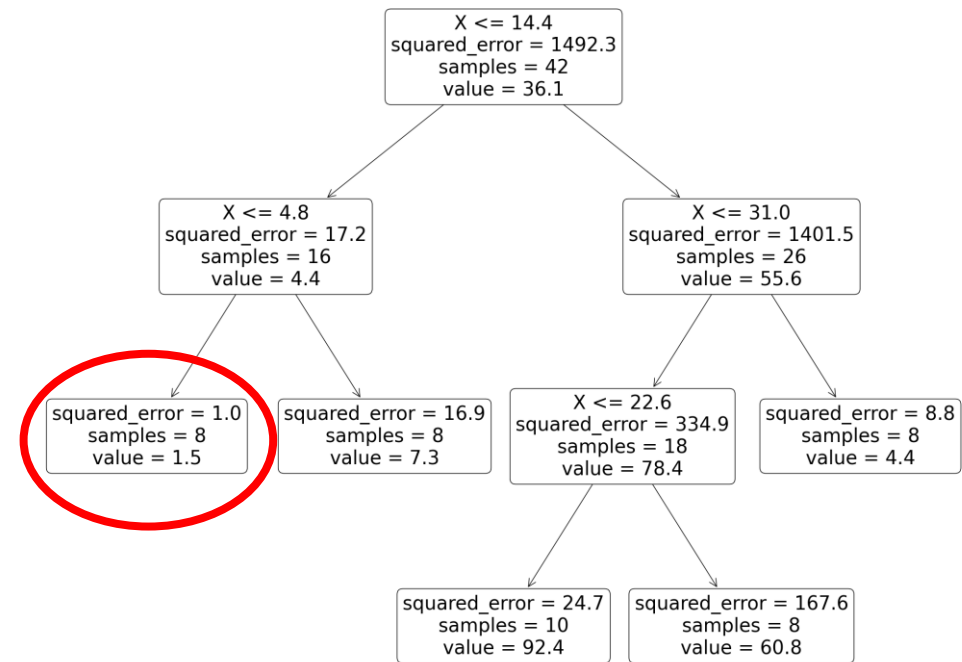
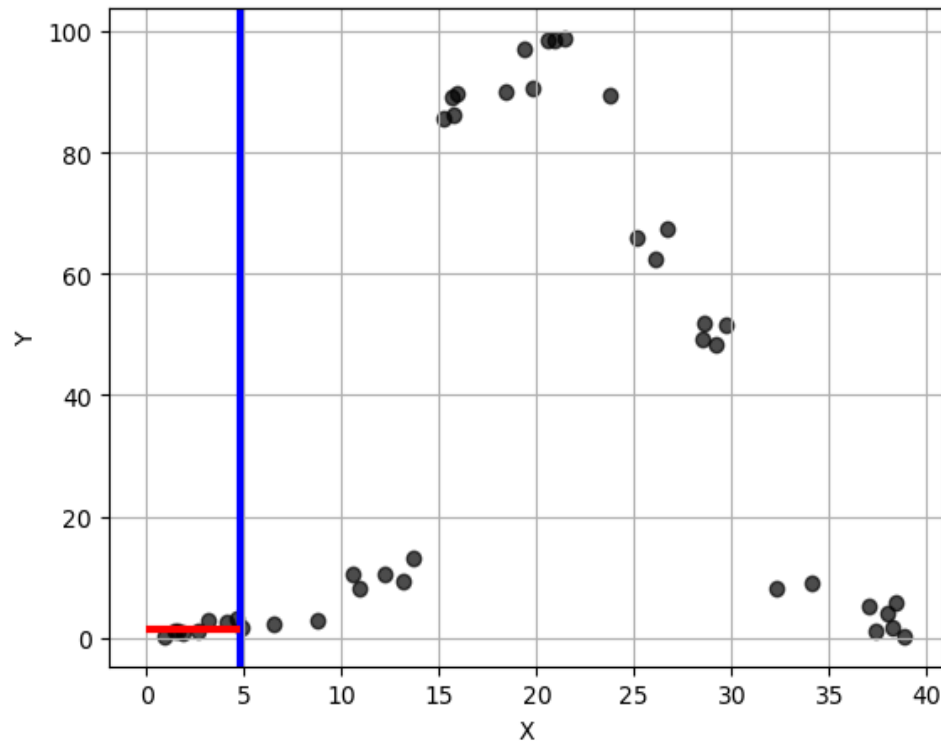


الگوریتم درخت تصمیم

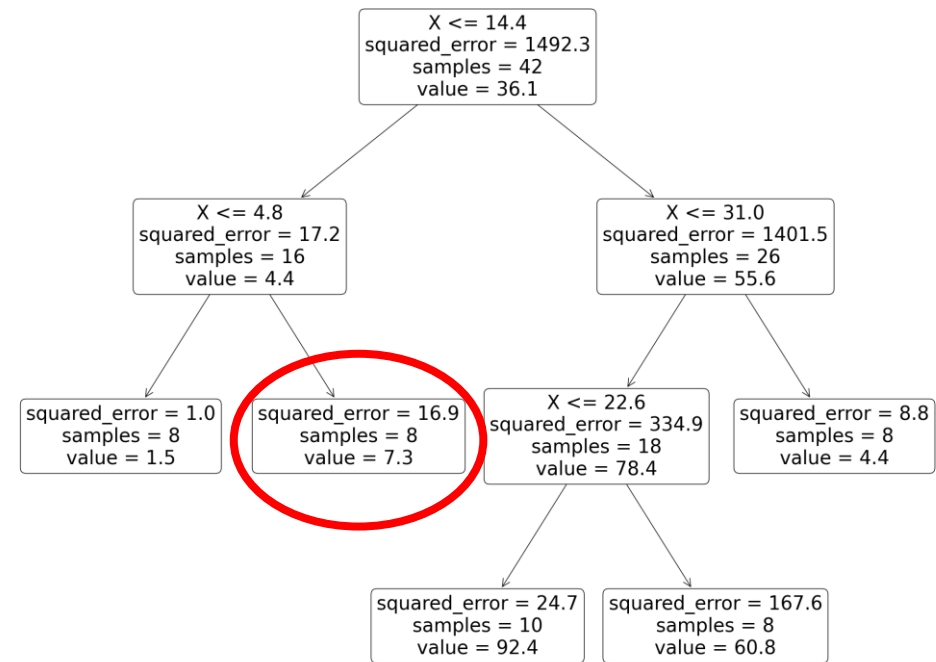
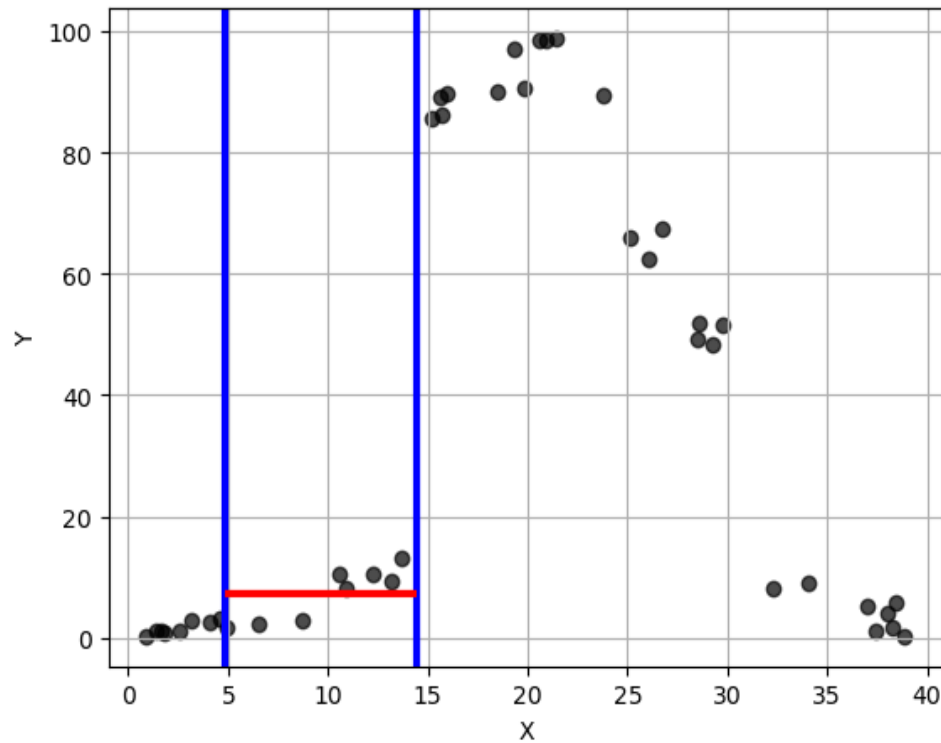
(Classification and Regression Tree - CART)



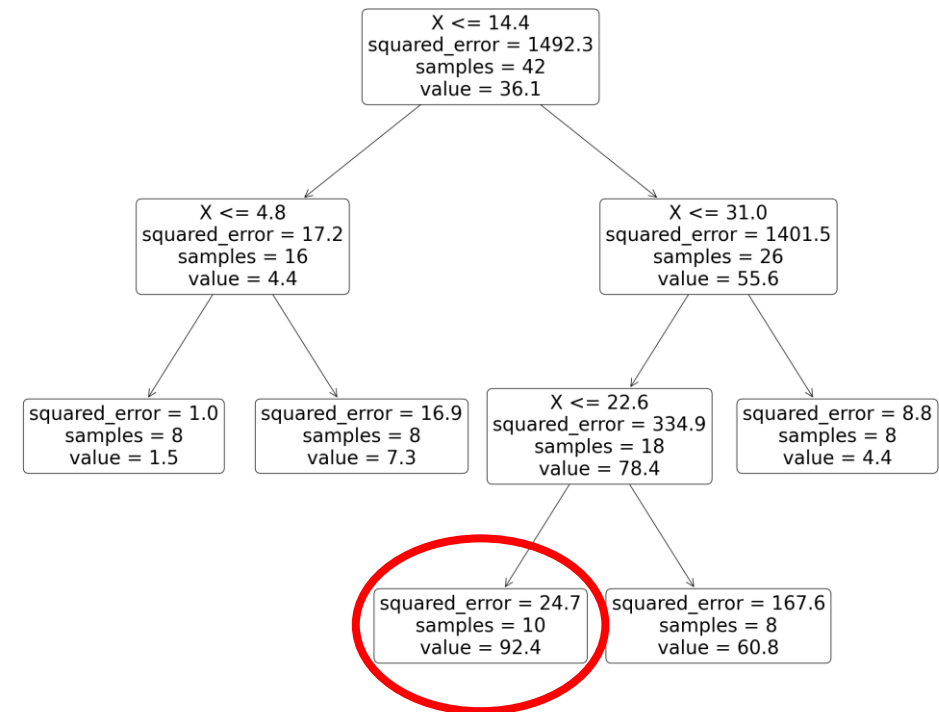
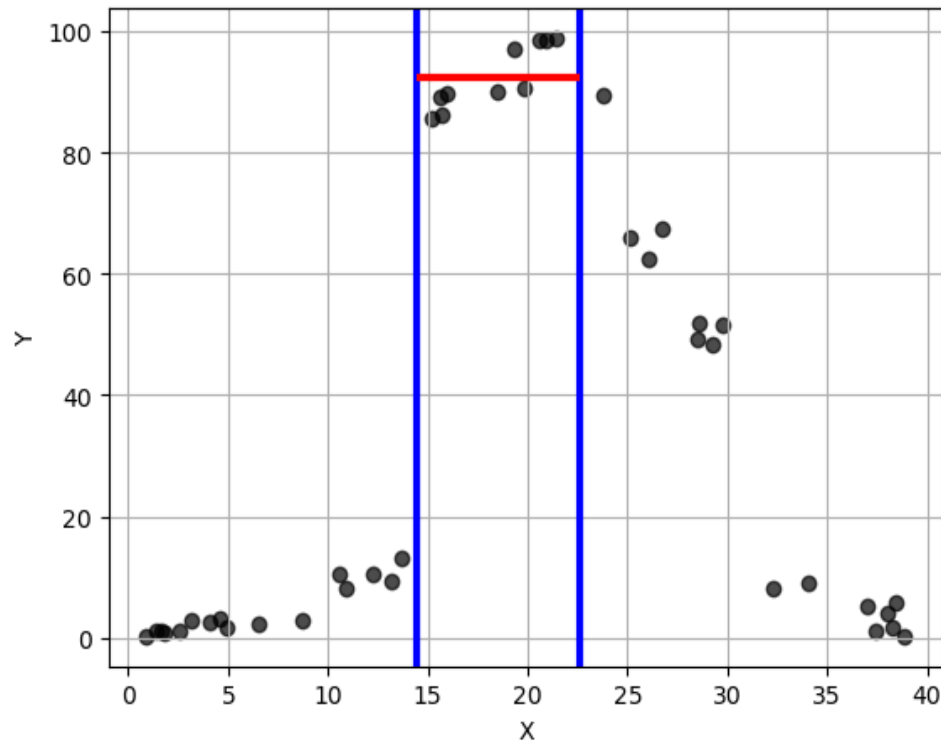
فهم شهودی الگوریتم درخت تصمیم



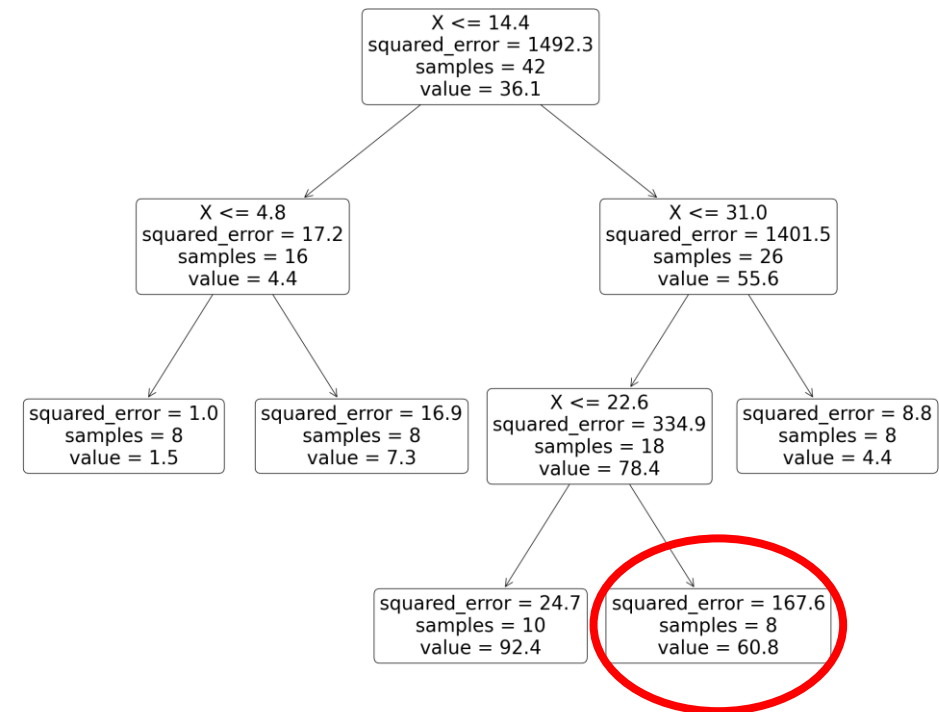
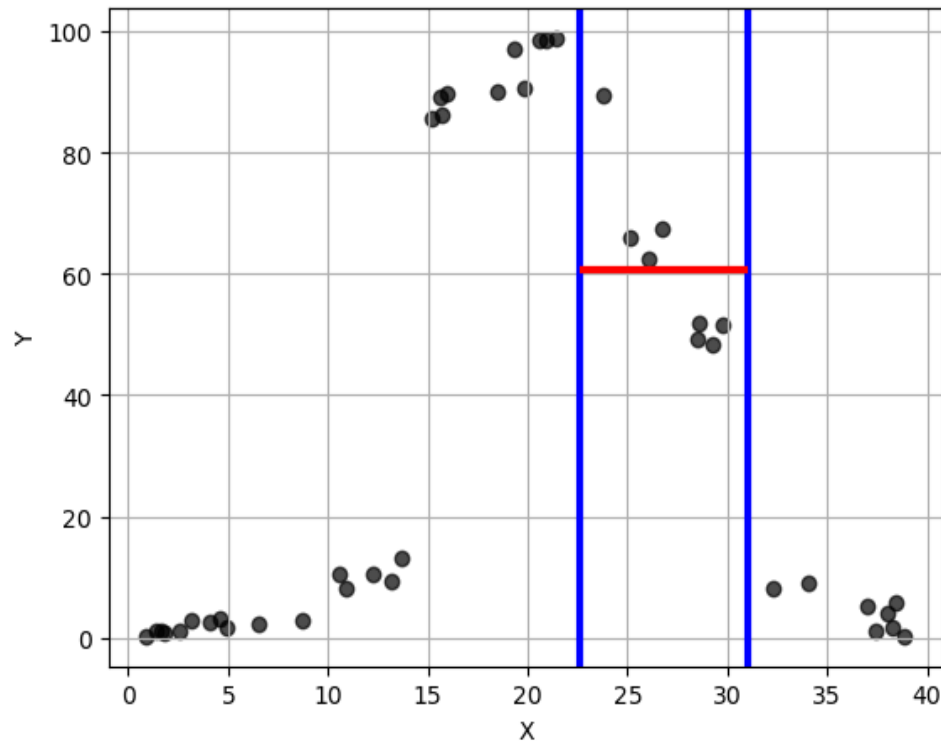
فهم شهودی الگوریتم درخت تصمیم



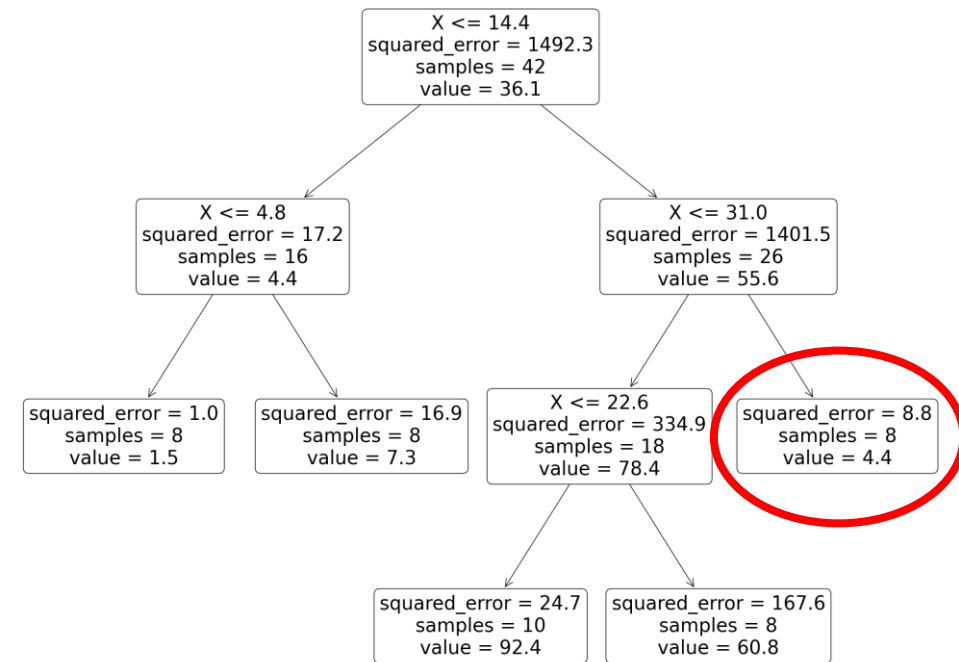
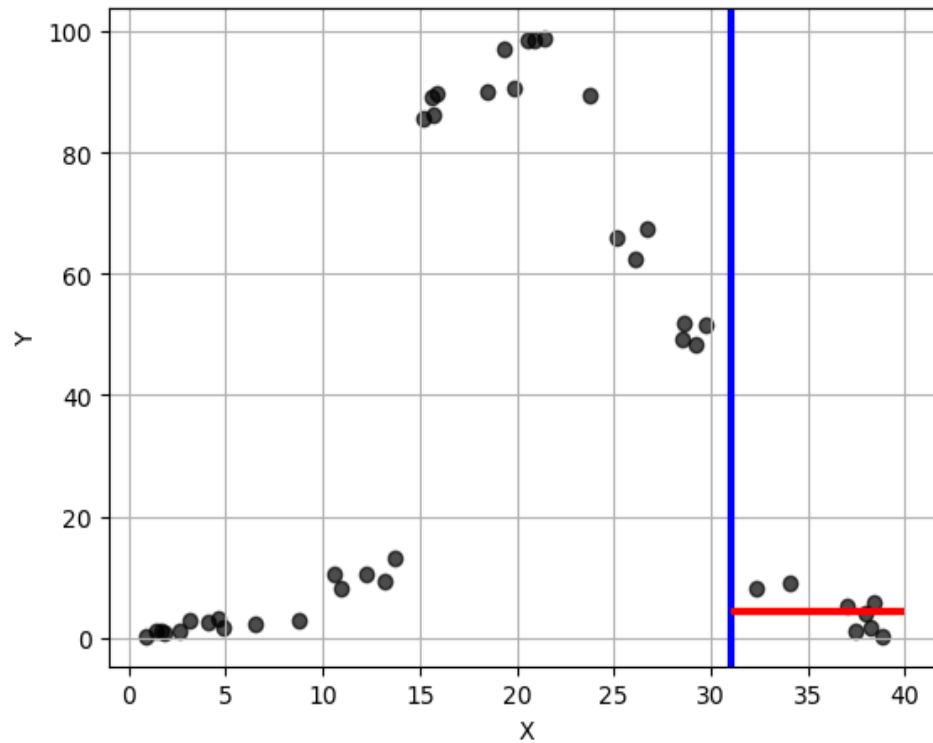
فهم شهودی الگوریتم درخت تصمیم



فهم شهودی الگوریتم درخت تصمیم



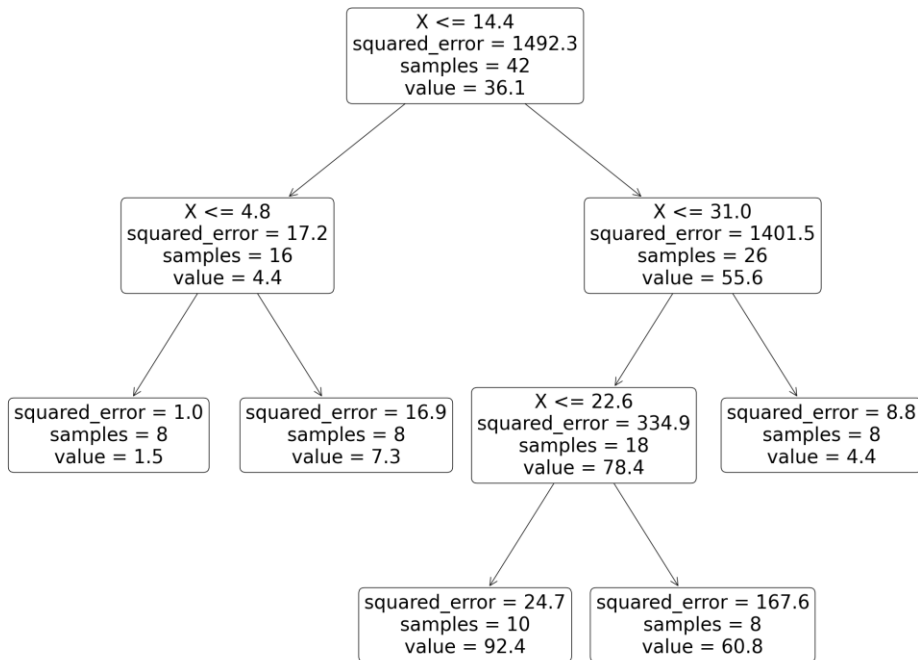
فهم شهودی الگوریتم درخت تصمیم



الگوریتم درخت تصمیم برای پیش‌بینی

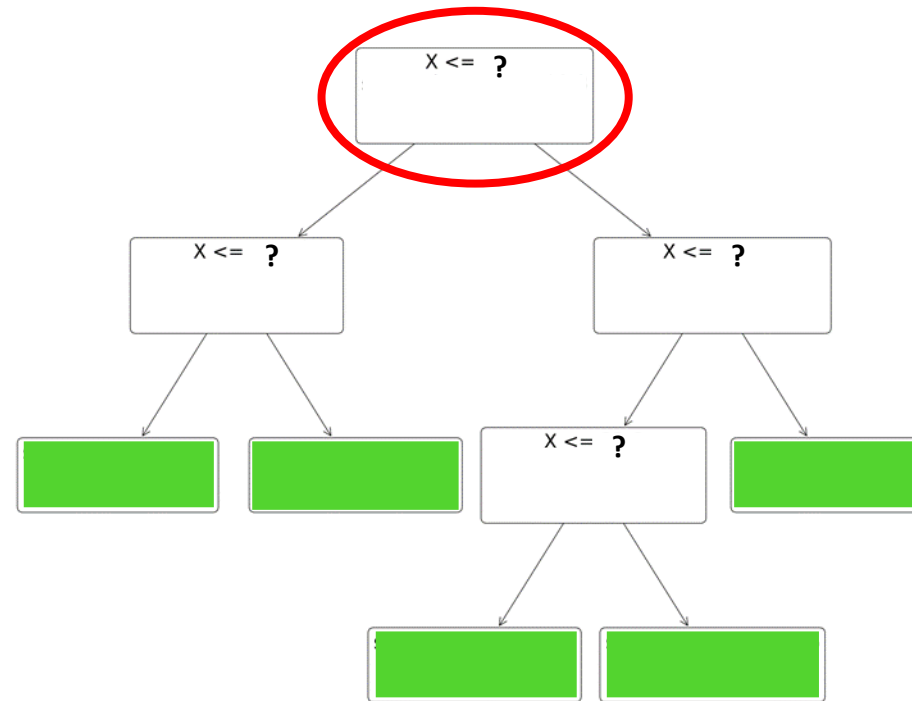
• پیش‌بینی یک مشاهده جدید

$$X = 20 \rightarrow y = ?$$



فهم شهودی الگوریتم درخت تصمیم

- چگونه آستانه هر گره مشخص می‌گردد؟



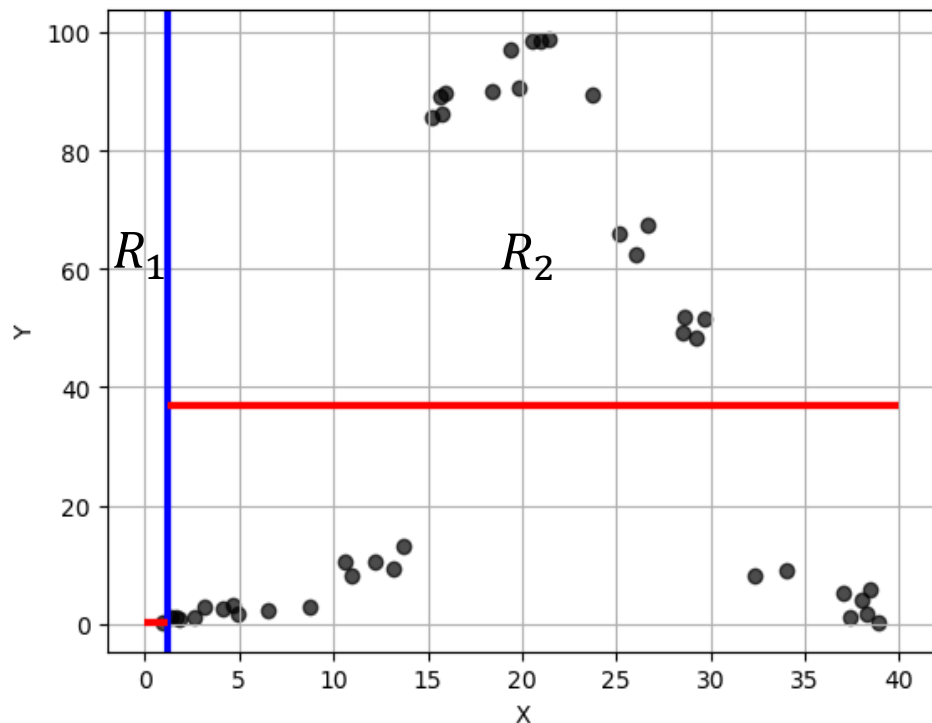
فهم شهودی الگوریتم درخت تصمیم

• چگونه آستانه هر گره مشخص می گردد؟

تابع هزینه (Cost Function)

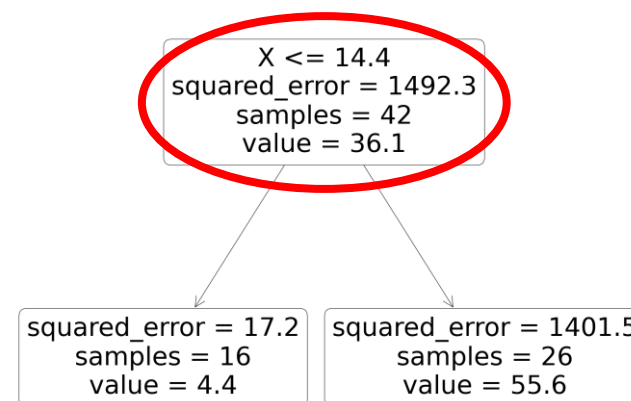
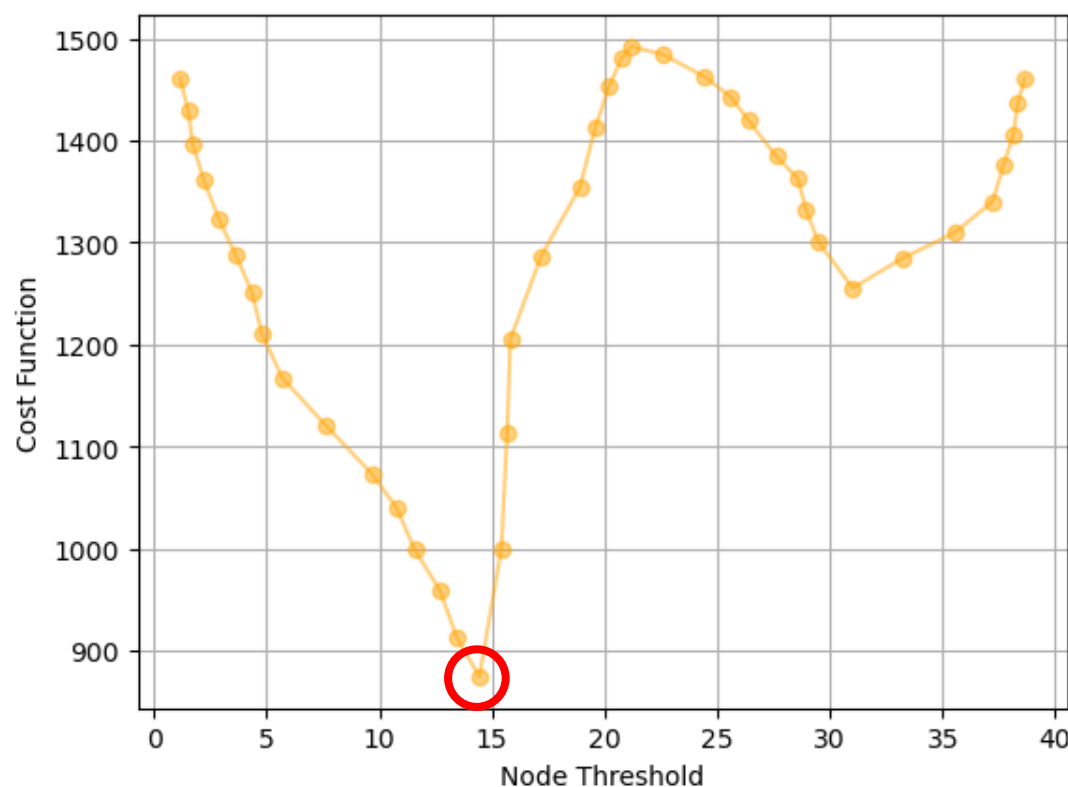
$$\text{Minimize: } \frac{1}{n} \left(\sum_{i \in R_1} (y_i - \hat{y}_{R_1})^2 + \sum_{i \in R_2} (y_i - \hat{y}_{R_2})^2 \right)$$

$$\text{Minimize: } \frac{n_{R_1}}{n} \text{MSE}_{R_1} + \frac{n_{R_2}}{n} \text{MSE}_{R_2}$$



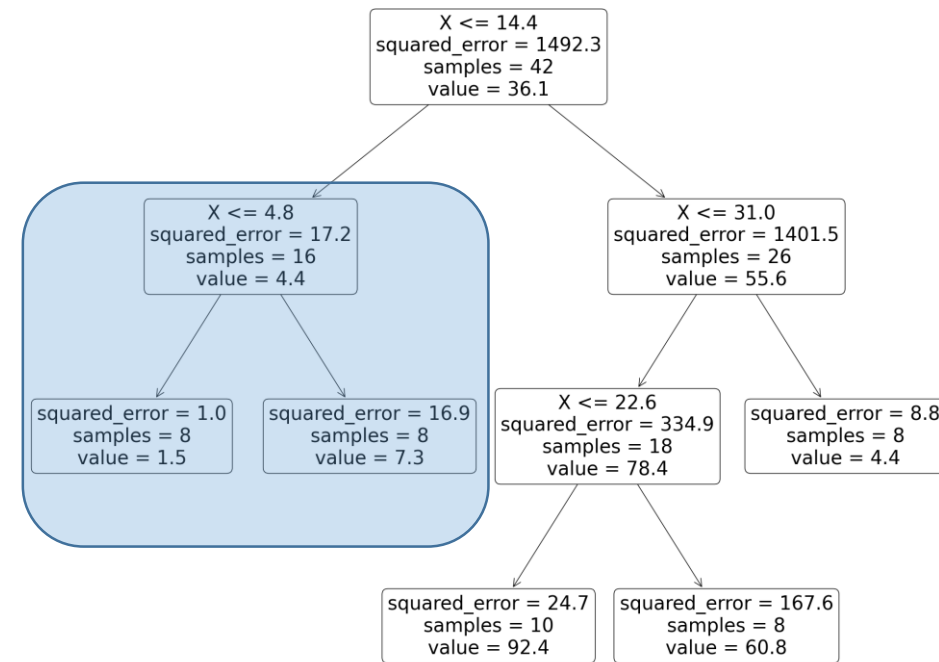
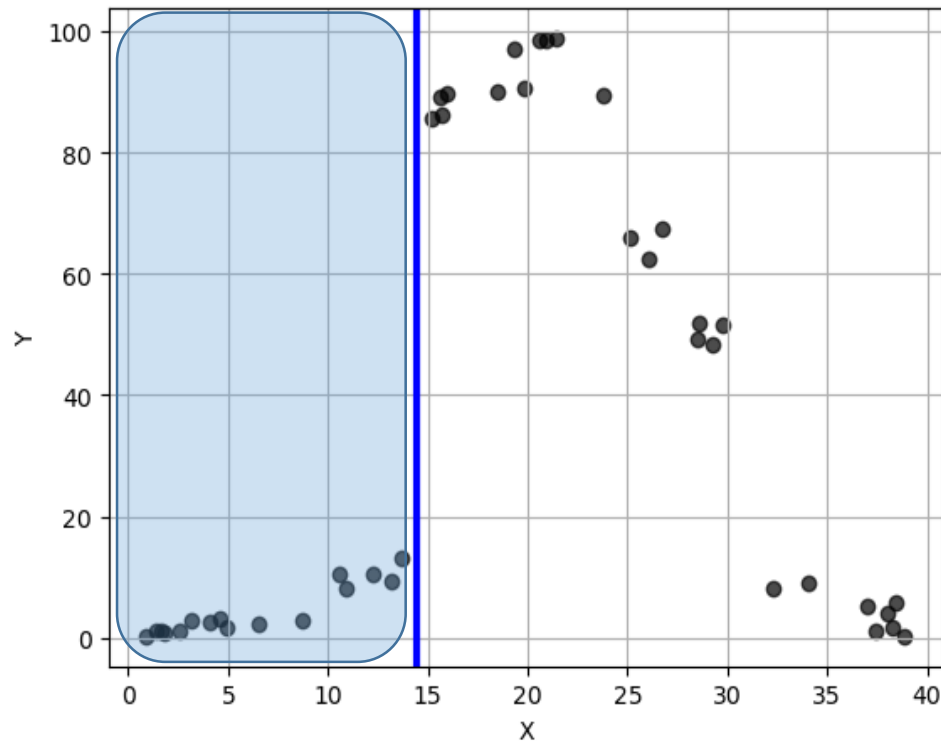
فهم شهودی الگوریتم درخت تصمیم

• چگونه آستانه هر گره مشخص می‌گردد؟



فهم شهودی الگوریتم درخت تصمیم

• چگونه آستانه هر گره مشخص می‌گردد؟



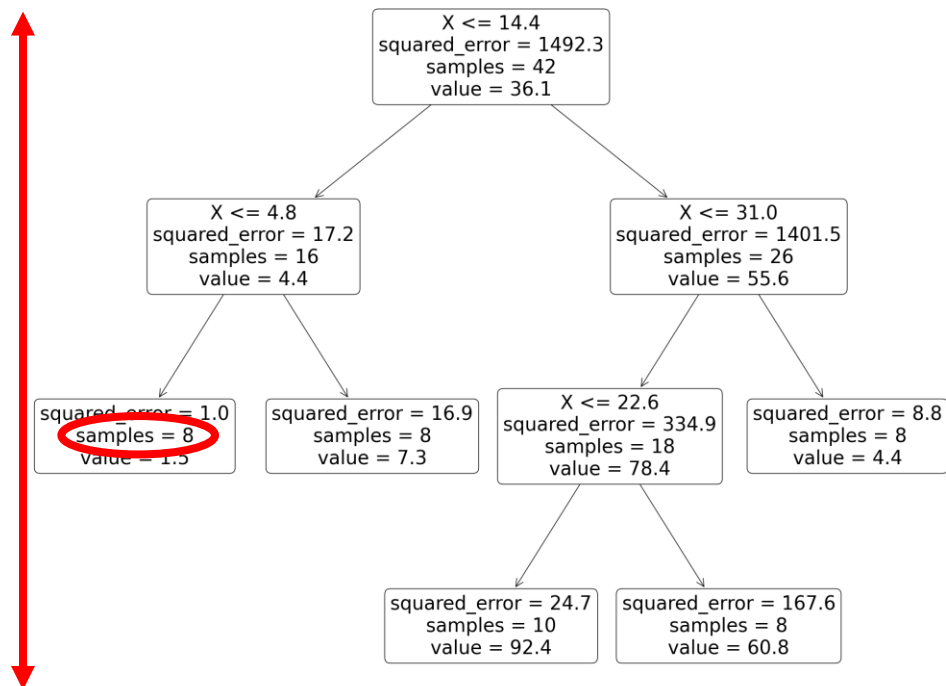
چگونه درخت متوقف می‌شود؟

• انتخاب فرایرامترها با استفاده از رویکرد

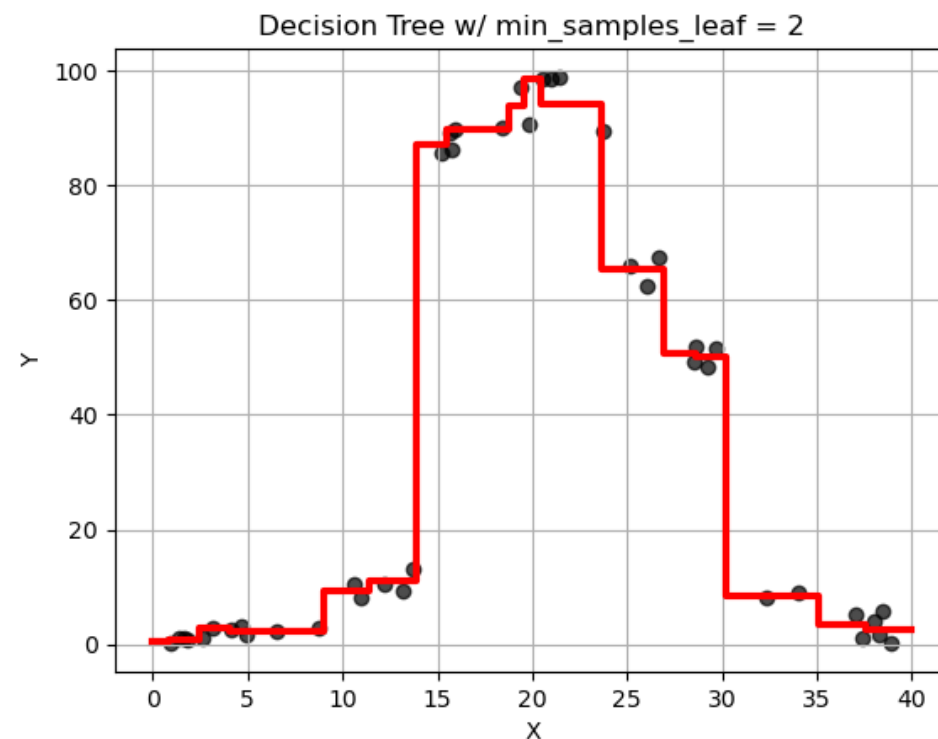
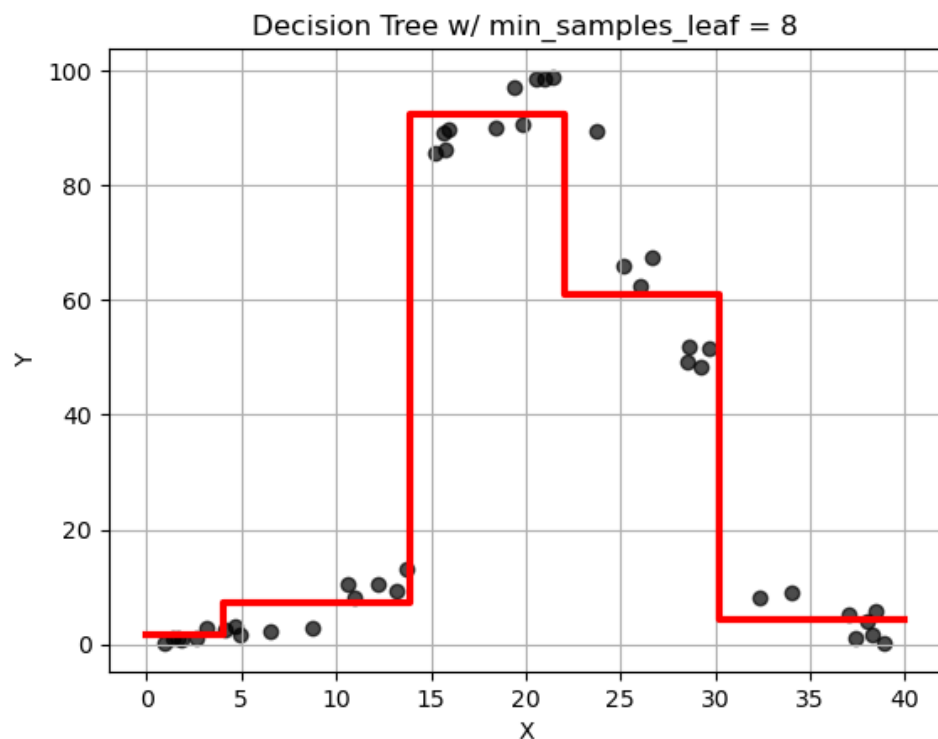
k-fold CV

• حداقل تعداد مشاهدات در هر برگ

• حداکثر عمق درخت

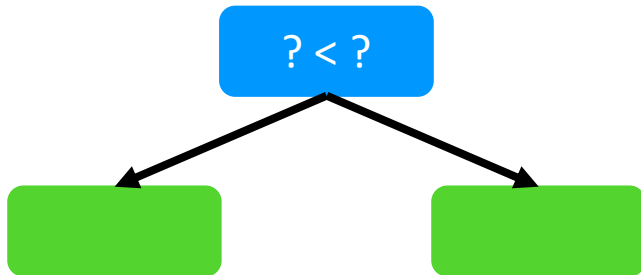


تأثیر فراپارامترها روی مدل



درخت تصمیم در حالت چندمتغیره

- فرض کنید مسئله‌ای با سه متغیر X_1 (عددی)، X_2 (عددی) و X_3 (رسته‌ای) داریم



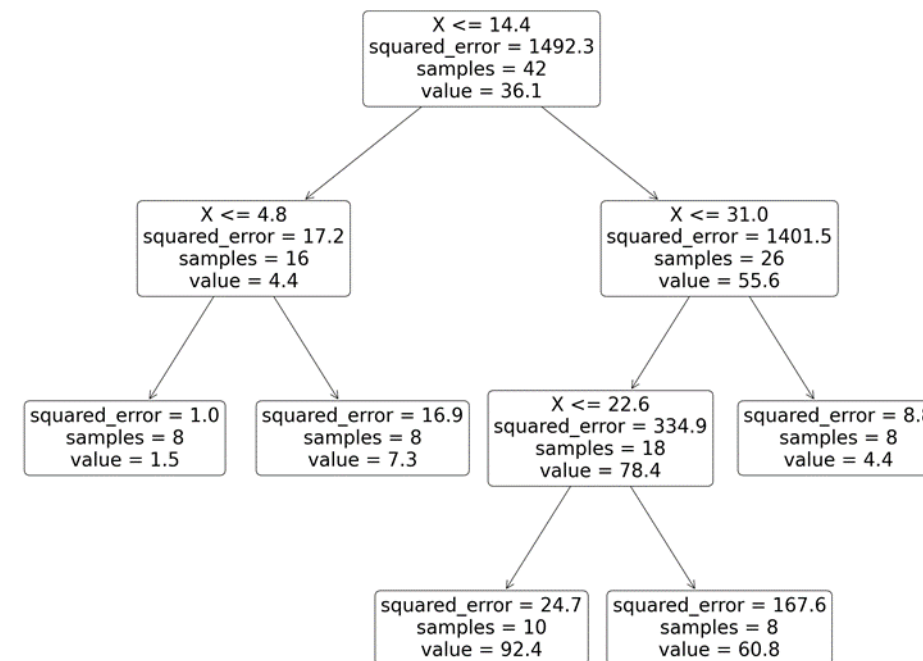
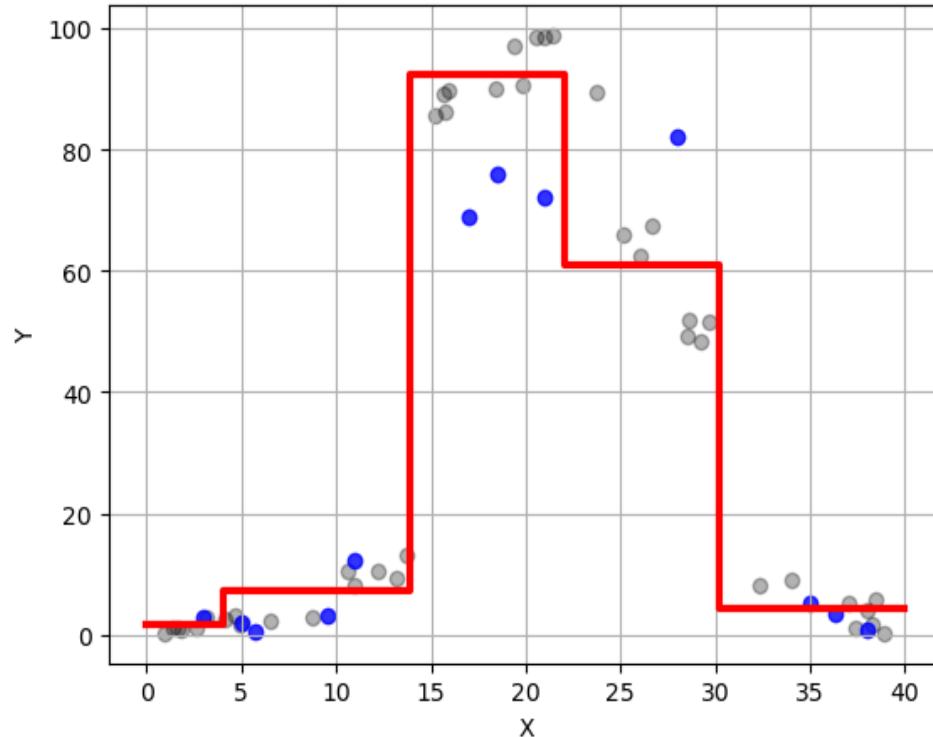
$$X_1 < 14.4 \rightarrow \text{Cost Function}_1$$

$$X_2 < 90.5 \rightarrow \text{Cost Function}_2$$

$$X_3 = A \rightarrow \text{Cost Function}_3$$

مفهوم حذف کردن شاخه‌ها در درخت تصمیم (Pruning)

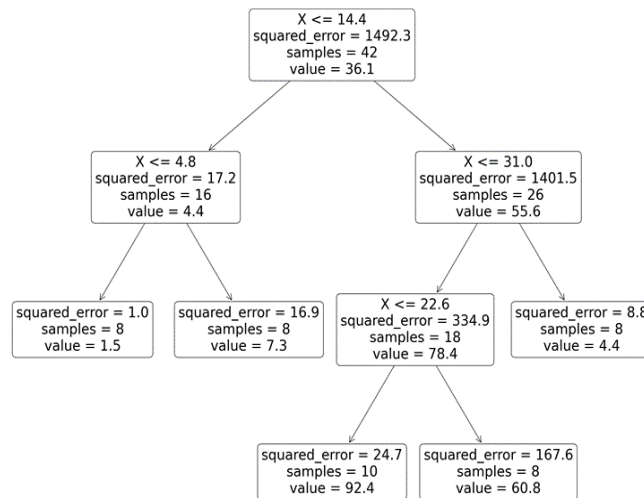
• عملکرد مدل روی داده‌های جدید (نقاط آبی رنگ) چگونه است؟



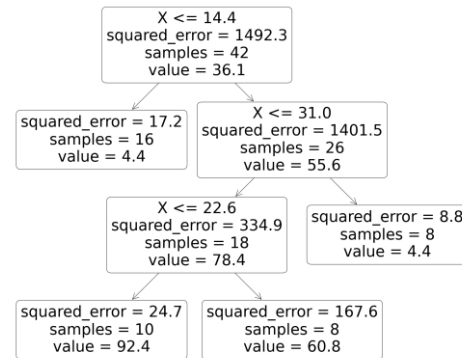
حذف کردن شاخه‌ها پس از ایجاد درخت اولیه (Post Pruning)

- انتخاب یک درخت زیرمجموعه‌ای از درخت پیچیده اولیه

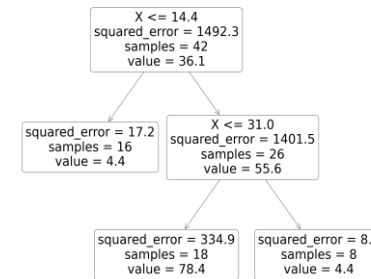
$$\text{Cost Complexity} = \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|, \quad \alpha \geq 0$$



$\alpha = 0$



$\alpha = 3.143$



$\alpha = 105.722$

squared_error = 1492.3
samples = 42
value = 36.1

$\alpha = 670.277$

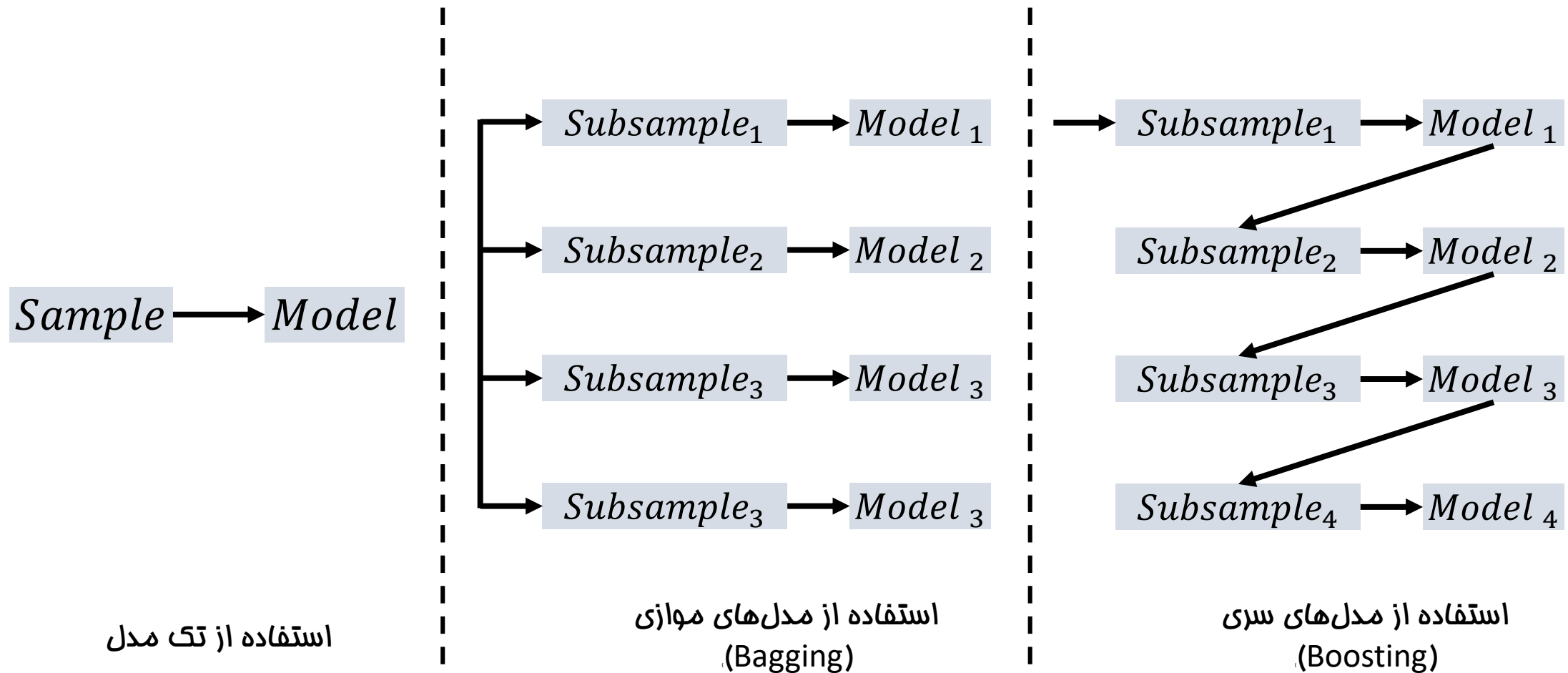
حذف کردن شاخه‌ها پس از ایجاد درخت اولیه

- گام اول: روی داده‌های آموزش با رویکرد جداسازی باینری تکرار شونده (به روشی که در کلاس بحث شد)، درخت تصمیم را ایجاد کنید. الگوریتم زمانی متوقف می‌گردد که تعداد مشاهدات در حداقل یک برگ از مقدار از قبل مشخص شده‌ای، کمتر گردد.
- گام دوم: با استفاده از رویکرد Cost Complexity Pruning، زیرمجموعه‌هایی از درخت پیچیده اولیه، تابعی از α ایجاد گردد.
- گام سوم: با استفاده از رویکرد k-Fold Cross Validation، بهترین مقدار α انتخاب گردد.
- گام چهارم: درخت زیرمجموعه معادل α انتخاب گردد.

چند نکته درباره الگوریتم درخت تصمیم

- الگوریتم درخت تصمیم برای رگرسیون، توسط کلاس `sklearn.tree.DecisionTreeRegressor` اجرا می‌شود.
- درخت تصمیم مدل ساده و قابل توضیحی را فراهم می‌کند.
- درخت تصمیم از الگوریتم‌هایی است که فاز آماده‌سازی چندان برای داده‌ها نیاز ندارد.
 - برای داده‌های رسته‌ای نیازی به تعریف متغیر کمکی نیست.
 - با مضمون داده‌های گم‌شده قابل اجرا است.
- زمان محاسبات برای آموزش مدل $O(n_{sample} \times \log(n_{sample}) \times m)$
- فروجهی درخت می‌تواند خیلی تحت تاثیر نمونه آموزش باشد (Non-robust).
- معمولاً یک درخت به تنهایی عملکرد فوق‌العاده‌ای ندارد.
 - درخت‌های پیچیده در معرض برازش بیش از حد (واریانس) هستند.
 - درخت‌های ساده در معرض سوگیری هستند.

ویکرد یادگیری جمعی (Ensemble Learning)



الگوریتم جنگل تصادفی (Random Forest)

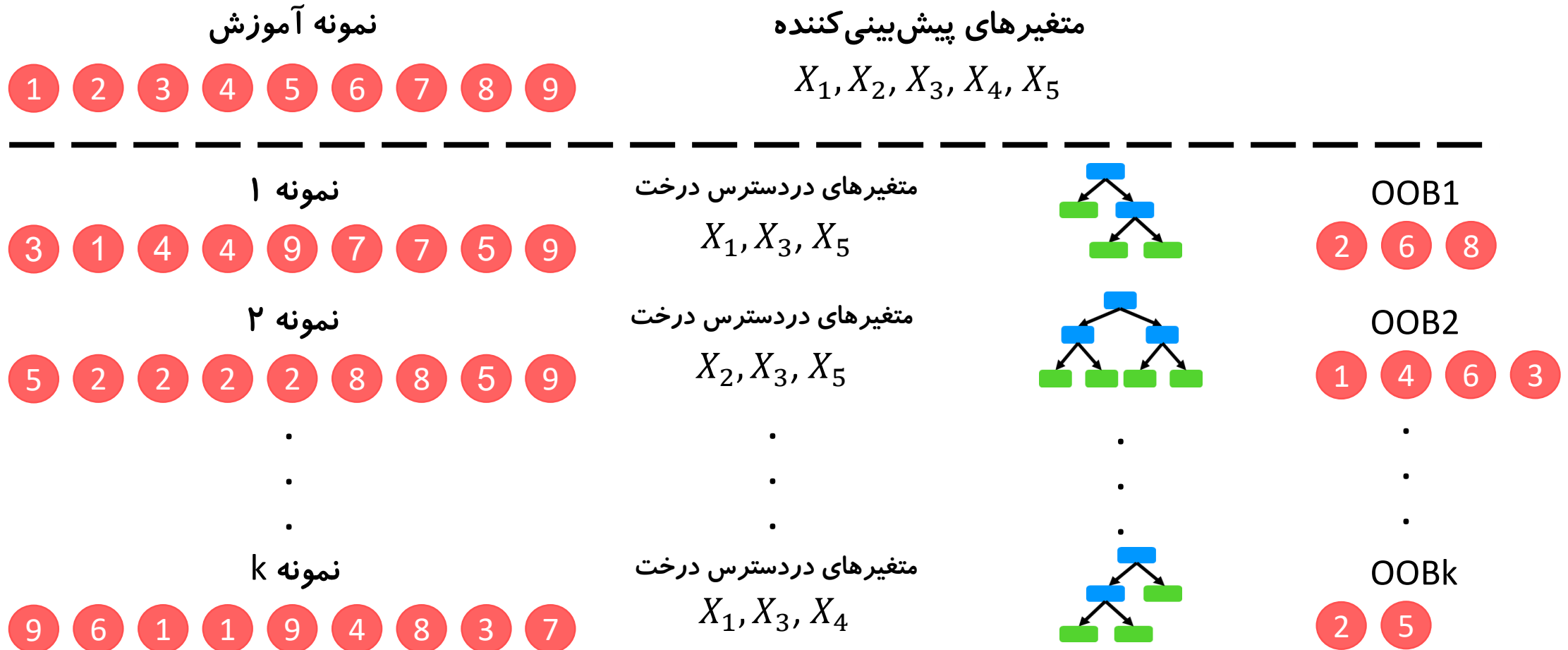
- استفاده از روش Bootstrap برای بازنمونه‌گیری
- انتخاب یک زیر مجموعه تصادفی از متغیرها ($\sqrt{\# \text{ of features}}$) در ساخت هر درخت
- پیش‌بینی بر اساس تجميع پیش‌بینی همه مدل‌ها

$$\hat{y}_{rf} = \text{Mean}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_k)$$

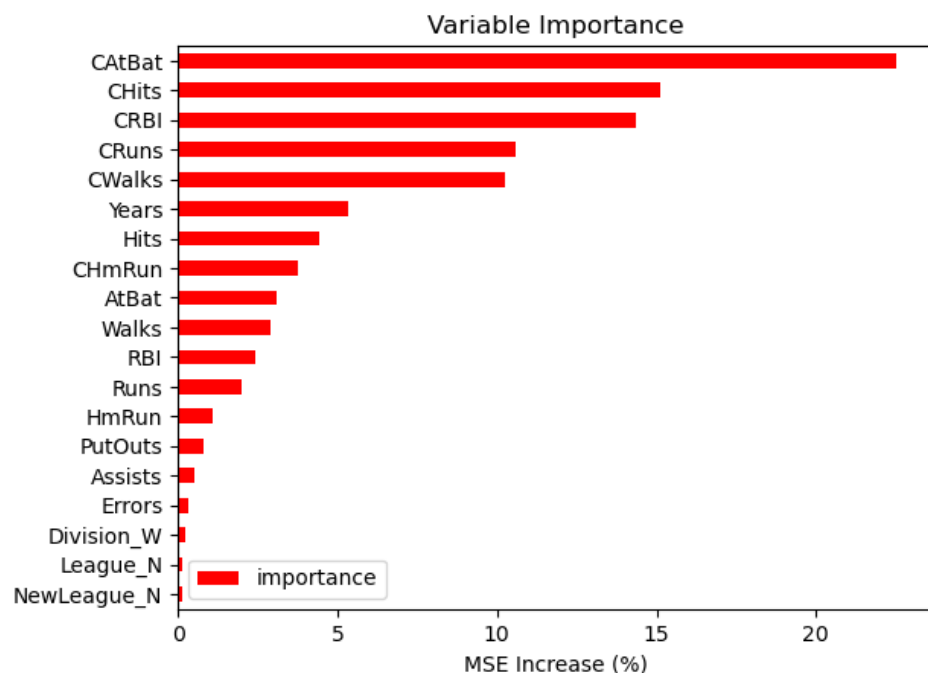
- الگوریتم جنگل تصادفی برای رگرسیون، توسط کلاس `sklearn.ensemble.RandomForestRegressor` اجرا می‌شود.

- زمان محاسبات برای آموزش مدل $O(\# \text{ of trees} \times n_{\text{sample}} \times \log(n_{\text{sample}}) \times m)$

فهم شهودی الگوریتم جنگل تصادفی



چگونه الگوریتم جنگل تصادفی را از حالت جعبه سیاه خارج کنیم؟



- محاسبه اهمیت متغیرها (Variable Importance)

- افزایش خطاها (MSE Increase)

- محاسبه خطای برآورد در OOB Sample

- میزان افزایش خطا در OOB در صورت دسترس نبودن یک

متغیر نسبت به حالتی که در دسترس درخت است

فهم شهودی الگوریتم تقویت گرادیان (Gradient Boost)

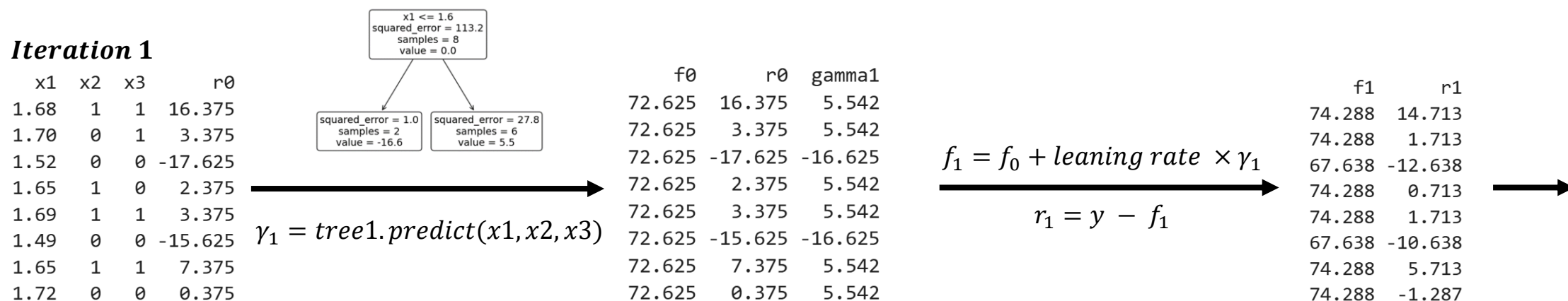
y	x1	x2	x3
89	1.68	1	1
76	1.70	0	1
55	1.52	0	0
75	1.65	1	0
76	1.69	1	1
57	1.49	0	0
80	1.65	1	1
73	1.72	0	0

فهم شهودی الگوریتم تقویت گرادیان

Iteration 0

y	x1	x2	x3		f0	r0
89	1.68	1	1		72.625	16.375
76	1.70	0	1		72.625	3.375
55	1.52	0	0	$f_0 = \text{mean}(y)$	72.625	-17.625
75	1.65	1	0	$r_0 = y - f_0$	72.625	2.375
76	1.69	1	1		72.625	3.375
57	1.49	0	0		72.625	-15.625
80	1.65	1	1		72.625	7.375
73	1.72	0	0		72.625	0.375

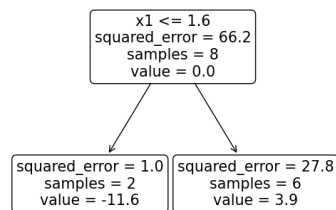
Iteration 1



فهم شهودی الگوریتم تقویت گرادیان

Iteration 2

x1	x2	x3	r1
1.68	1	1	14.7125
1.70	0	1	1.7125
1.52	0	0	-12.6375
1.65	1	0	0.7125
1.69	1	1	1.7125
1.49	0	0	-10.6375
1.65	1	1	5.7125
1.72	0	0	-1.2875



$\gamma_2 = \text{tree2.predict}(x1, x2, x3)$

f1	r1	gamma2
74.288	14.713	3.879
74.288	1.713	3.879
67.638	-12.638	-11.638
74.288	0.713	3.879
74.288	1.713	3.879
67.638	-10.638	-11.638
74.288	5.713	3.879
74.288	-1.287	3.879

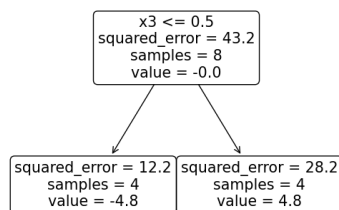
$$f_2 = f_1 + \text{leaning rate} \times \gamma_2$$

$$r_2 = y - f_2$$

f2	r2
75.451	13.549
75.451	0.549
64.146	-9.146
75.451	-0.451
75.451	0.549
64.146	-7.146
75.451	4.549
75.451	-2.451

Iteration 3

x1	x2	x3	r2
1.68	1	1	13.54875
1.70	0	1	0.54875
1.52	0	0	-9.14625
1.65	1	0	-0.45125
1.69	1	1	0.54875
1.49	0	0	-7.14625
1.65	1	1	4.54875
1.72	0	0	-2.45125



$\gamma_3 = \text{tree3.predict}(x1, x2, x3)$

f2	r2	gamma3
75.451	13.549	4.799
75.451	0.549	4.799
64.146	-9.146	-4.799
75.451	-0.451	-4.799
75.451	0.549	4.799
64.146	-7.146	-4.799
75.451	4.549	4.799
75.451	-2.451	-4.799

$$f_3 = f_2 + \text{leaning rate} \times \gamma_3$$

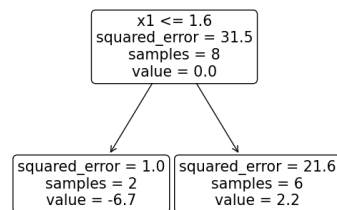
$$r_3 = y - f_3$$

f3	r3
76.891	12.109
76.891	-0.891
62.707	-7.707
74.012	0.988
76.891	-0.891
62.707	-5.707
76.891	3.109
74.012	-1.012

فهم شهودی الگوریتم تقویت گرادیان

Iteration 4

x1	x2	x3	r3
1.68	1	1	12.109
1.70	0	1	-0.891
1.52	0	0	-7.707
1.65	1	0	0.988
1.69	1	1	-0.891
1.49	0	0	-5.707
1.65	1	1	3.109
1.72	0	0	-1.012



$\gamma_4 = \text{tree4.predict}(x1, x2, x3)$

f3	r3	gamma4
76.891	12.109	2.236
76.891	-0.891	2.236
62.707	-7.707	-6.707
74.012	0.988	2.236
76.891	-0.891	2.236
62.707	-5.707	-6.707
76.891	3.109	2.236
74.012	-1.012	2.236

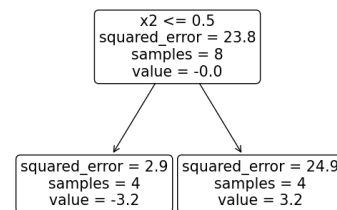
$$f_4 = f_3 + \text{leaning rate} \times \gamma_4$$

$$r_4 = y - f_4$$

f4	r4
77.562	11.438
77.562	-1.562
60.695	-5.695
74.682	0.318
77.562	-1.562
60.695	-3.695
77.562	2.438
74.682	-1.682

Iteration 5

x1	x2	x3	r4
1.68	1	1	11.438
1.70	0	1	-1.562
1.52	0	0	-5.695
1.65	1	0	0.318
1.69	1	1	-1.562
1.49	0	0	-3.695
1.65	1	1	2.438
1.72	0	0	-1.682



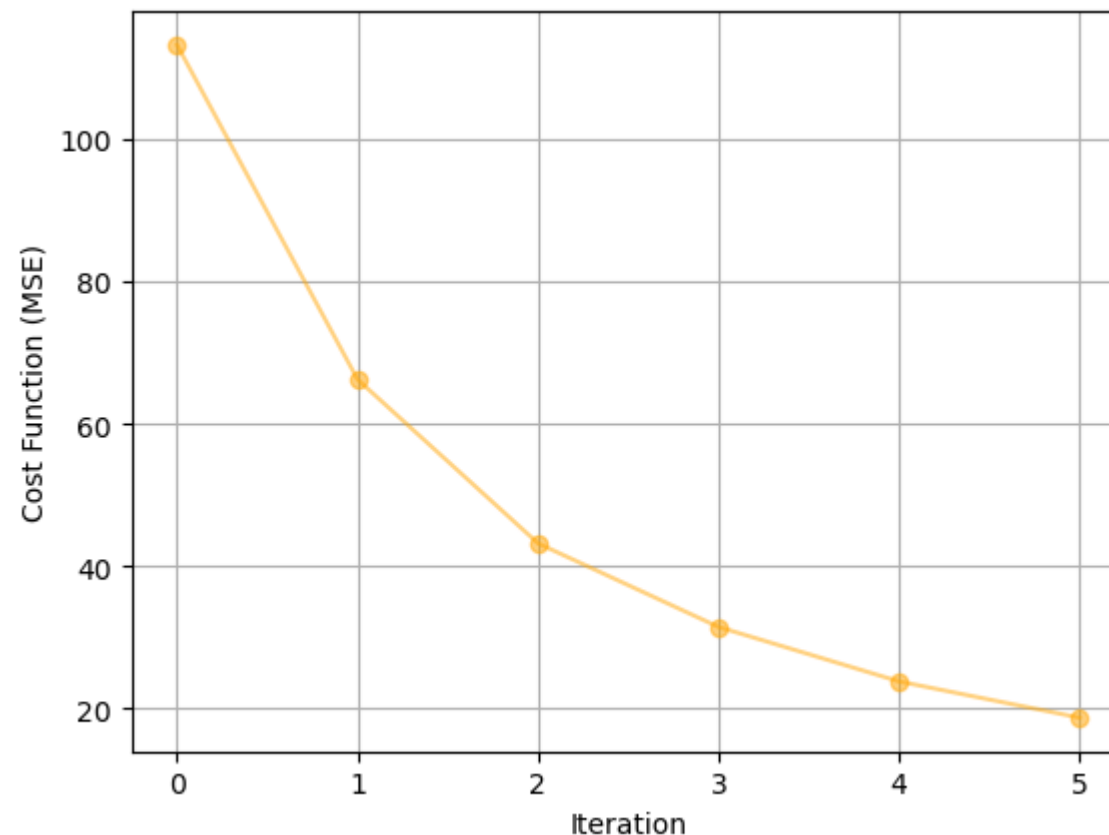
$\gamma_5 = \text{tree5.predict}(x1, x2, x3)$

f4	r4	gamma5
77.562	11.438	3.158
77.562	-1.562	-3.158
60.695	-5.695	-3.158
74.682	0.318	3.158
77.562	-1.562	3.158
60.695	-3.695	-3.158
77.562	2.438	3.158
74.682	-1.682	-3.158

$$f_5 = f_4 + \text{leaning rate} \times \gamma_5$$

y	x1	x2	x3	f5
89	1.68	1	1	78.509
76	1.70	0	1	76.615
55	1.52	0	0	59.748
75	1.65	1	0	75.629
76	1.69	1	1	78.509
57	1.49	0	0	59.748
80	1.65	1	1	78.509
73	1.72	0	0	73.735

تغییرات تابع هزینه در تکرارهای مختلف



پیشنهاد مطالعه



Stochastic Gradient Boosting

Jerome H. Friedman*

March 26, 1999

Abstract

Gradient boosting constructs additive regression models by sequentially fitting a simple parameterized function (base learner) to current “pseudo”-residuals by least-squares at each iteration. The pseudo-residuals are the gradient of the loss functional being minimized, with respect to the model values at each training data point, evaluated at the current step. It is shown that both the approximation accuracy and execution speed of gradient boosting can be substantially improved by incorporating randomization into the procedure. Specifically, at each iteration a subsample of the training data is drawn at random (without replacement) from the full training data set. This randomly selected subsample is then used in place of the full sample to fit the base learner and compute the model update for the current iteration. This randomized approach also increases robustness against overcapacity of the base learner.

Gradient Boosting Algorithm

```
1  $F_0(\mathbf{x}) = \arg \min_{\gamma} \sum_{i=1}^N \Psi(y_i, \gamma)$   
2 For  $m = 1$  to  $M$  do:  
3    $\tilde{y}_{im} = - \left[ \frac{\partial \Psi(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)} \right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, i = 1, N$   
4    $\{R_{lm}\}_1^L = L - \text{terminal node } tree(\{\tilde{y}_{im}, \mathbf{x}_i\}_1^N)$   
5    $\gamma_{lm} = \arg \min_{\gamma} \sum_{\mathbf{x}_i \in R_{lm}} \Psi(y_i, F_{m-1}(\mathbf{x}_i) + \gamma)$   
6    $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot \gamma_{lm} 1(\mathbf{x} \in R_{lm})$   
7 endFor
```

الگوریتم Stochastic Gradient Boost

- در ساخت هر درخت

- زیرمجموعه‌ای از مشاهدات (بین ۵۰ تا ۸۰ درصد مشاهدات) استفاده شود.

- زیرمجموعه‌ای از متغیرهای پیش‌بینی‌کننده استفاده گردد.

- بویژه در حضور هم‌خطی و تعداد بالای متغیرهای پیش‌بینی‌کننده

چند نکته درباره الگوریتم تقویت گرادیان

- الگوریتم تقویت گرادیان برای رگرسیون، توسط کلاس `sklearn.ensemble.GradientBoostingRegressor` اجرا می‌شود.
- زمان محاسبات برای آموزش مدل $O(\# \text{ of trees} \times n_{\text{sample}} \times \log(n_{\text{sample}}) \times m)$
- مدل‌ساز درباره فرآیندهای زیر تصمیم می‌گیرد
 - نرخ یادگیری (بین 0.001 تا 0.3)
 - درصدی از مشاهدات و متغیرهای پیش‌بینی‌کننده که در سافت هر درخت استفاده می‌شود
 - تعداد درختان (معمولاً خیلی زیاد نیست)
 - بیشینه عمق درخت (بین ۱ تا ۵)
 - کمینه تعداد مشاهدات در هر برگ درخت

برنامه‌نویسی در پایتون

موردکاوی الگوریتم درخت تصمیم،
جنگل تصادفی و تقویت گرادیان



