COMP 4432 Machine Learning

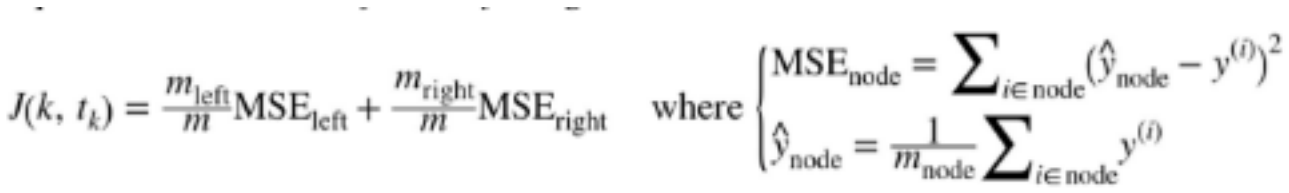
Lesson 6: Decision Trees

Agenda

- Questions on Assignment 3
- Regression Tree Refresher
- Classification Tree
- Complexity
- Feature Importance

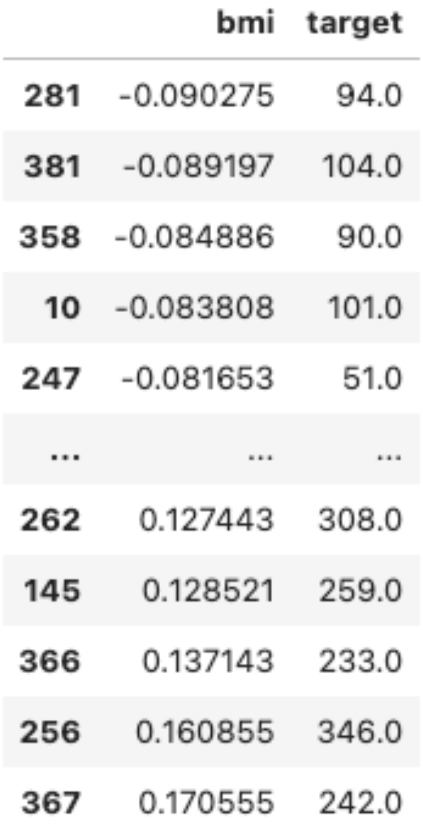
Assignment 3 Questions

Decision Tree Regressor



$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Data is sorted
by feature



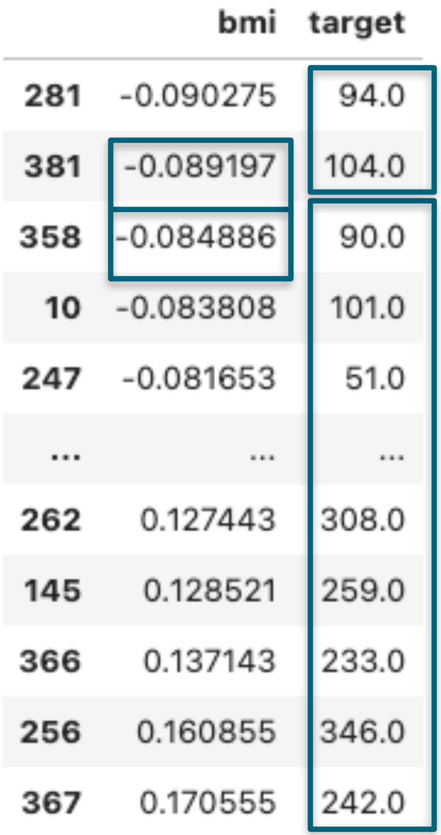
	bmi	target
281	-0.090275	94.0
381	-0.089197	104.0
358	-0.084886	90.0
10	-0.083808	101.0
247	-0.081653	51.0
...	...	...
262	0.127443	308.0
145	0.128521	259.0
366	0.137143	233.0
256	0.160855	346.0
367	0.170555	242.0

Decision Tree Regressor

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Data is sorted by feature

Calculate Mean as Cut Point

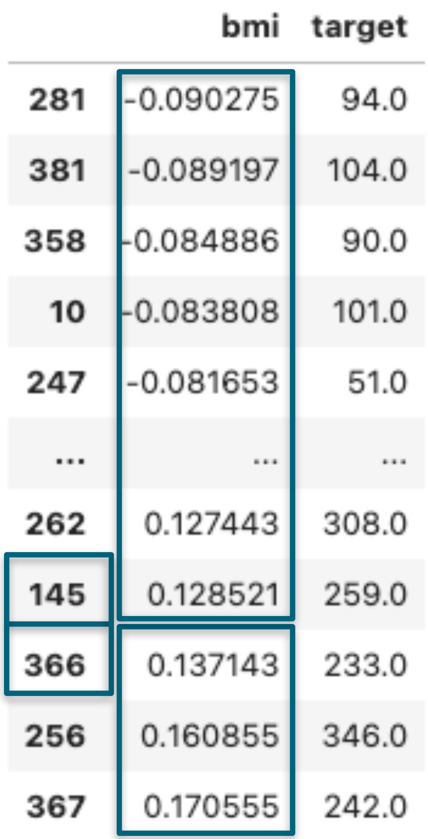


	bmi	target
281	-0.090275	94.0
381	-0.089197	104.0
358	-0.084886	90.0
10	-0.083808	101.0
247	-0.081653	51.0
...	...	...
262	0.127443	308.0
145	0.128521	259.0
366	0.137143	233.0
256	0.160855	346.0
367	0.170555	242.0

Calculate Mean Squared Error for each set Left & Right

Decision Tree Regressor

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$



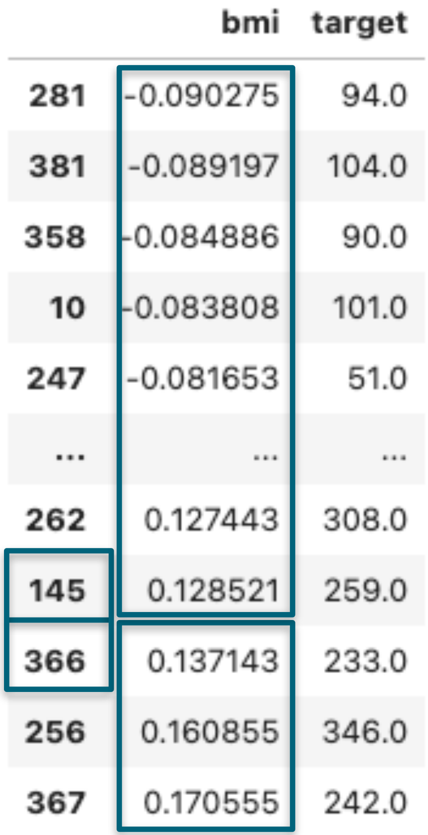
	bmi	target
281	-0.090275	94.0
381	-0.089197	104.0
358	-0.084886	90.0
10	-0.083808	101.0
247	-0.081653	51.0
...	...	...
262	0.127443	308.0
145	0.128521	259.0
366	0.137143	233.0
256	0.160855	346.0
367	0.170555	242.0

Calculate Mean as Cut Point

Calculate Mean Squared Error for each set Left & Right

Decision Tree Regressor

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$



	bmi	target
281	-0.090275	94.0
381	-0.089197	104.0
358	-0.084886	90.0
10	-0.083808	101.0
247	-0.081653	51.0
...	...	...
262	0.127443	308.0
145	0.128521	259.0
366	0.137143	233.0
256	0.160855	346.0
367	0.170555	242.0

Calculate Mean as Cut Point

Calculate Mean Squared Error for each set Left & Right

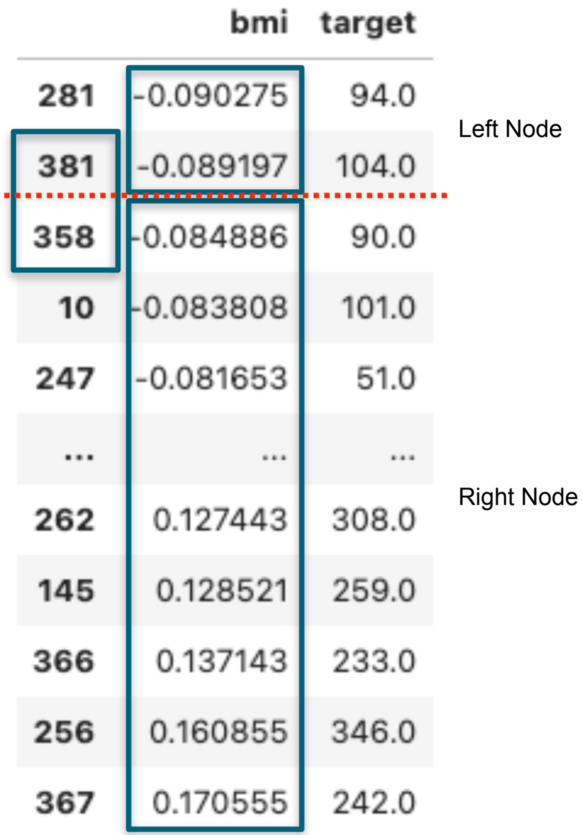
Decision Tree Regressor

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} (\hat{y}_{\text{node}} - y^{(i)})^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Calculate the midpoint between i and i + 1 (split point)

Unique values of feature

Calculate the MSE for the target values on either side of split



	bmi	target
281	-0.090275	94.0
381	-0.089197	104.0
358	-0.084886	90.0
10	-0.083808	101.0
247	-0.081653	51.0
...	...	...
262	0.127443	308.0
145	0.128521	259.0
366	0.137143	233.0
256	0.160855	346.0
367	0.170555	242.0

Left Node

Right Node

Decision Tree Regressor

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

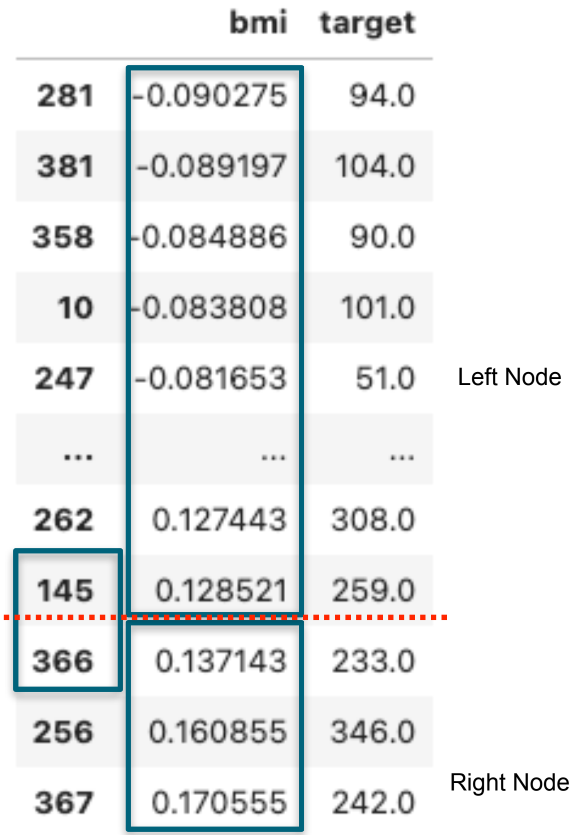
Calculate the midpoint between i and i + 1 (split point)

  Unique values of feature

Calculate the MSE for the target values on either side of split

Calculate the Total Cost

Gives (Split Point, Total Cost)



	bmi	target
281	-0.090275	94.0
381	-0.089197	104.0
358	-0.084886	90.0
10	-0.083808	101.0
247	-0.081653	51.0
...	...	...
262	0.127443	308.0
145	0.128521	259.0
366	0.137143	233.0
256	0.160855	346.0
367	0.170555	242.0

Left Node
Right Node

Decision Tree Regressor

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} (\hat{y}_{\text{node}} - y^{(i)})^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Calculate the midpoint between i and i + 1 (split point)

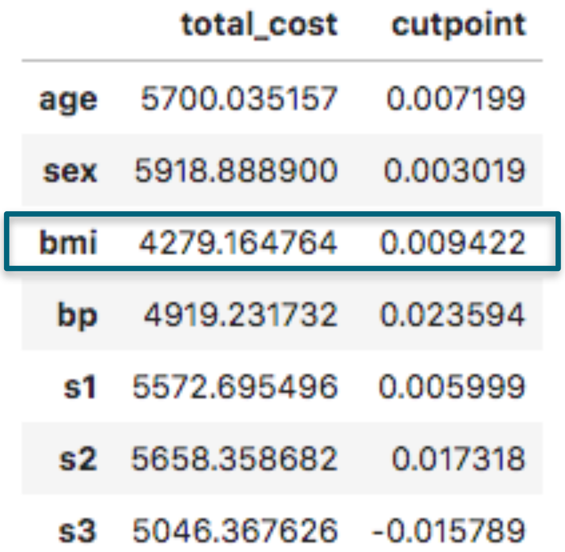
Unique values of feature

Calculate the MSE for the target values on either side of split

Calculate the Total Cost

Gives (Split Point, Total Cost)

Identify the Split Point with the Minimum Total Cost per Feature



	total_cost	cutpoint
age	5700.035157	0.007199
sex	5918.888900	0.003019
bmi	4279.164764	0.009422
bp	4919.231732	0.023594
s1	5572.695496	0.005999
s2	5658.358682	0.017318
s3	5046.367626	-0.015789

Decision Tree Regressor

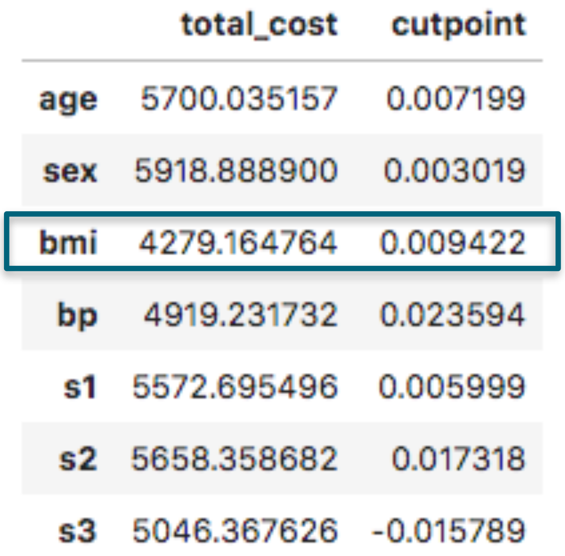
$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} (\hat{y}_{\text{node}} - y^{(i)})^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Calculate the midpoint between i and i + 1 (split point)
    Unique values of feature
Calculate the MSE for the target values on either side of split
Calculate the Total Cost
Gives (Split Point, Total Cost)

Identify the Split Point with the Minimum Total Cost per Feature

If multiple features are being considered, each feature follows the method above giving :
(Feature, Optimal Split Point, Minimum Total Cost)

The feature with the most minimum Total Cost is selected.

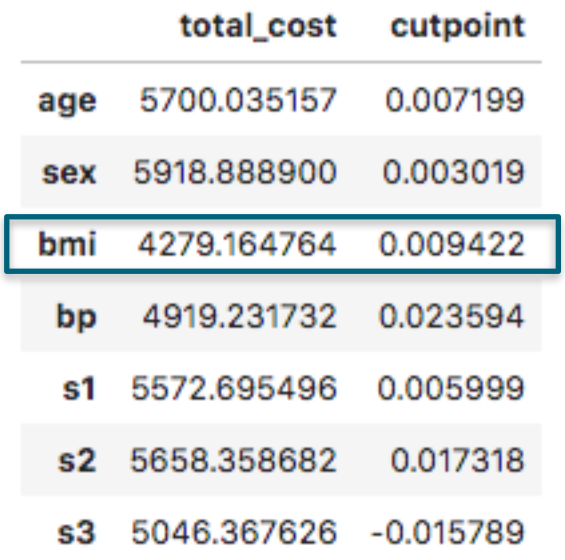


	total_cost	cutpoint
age	5700.035157	0.007199
sex	5918.888900	0.003019
bmi	4279.164764	0.009422
bp	4919.231732	0.023594
s1	5572.695496	0.005999
s2	5658.358682	0.017318
s3	5046.367626	-0.015789

Decision Tree Regressor

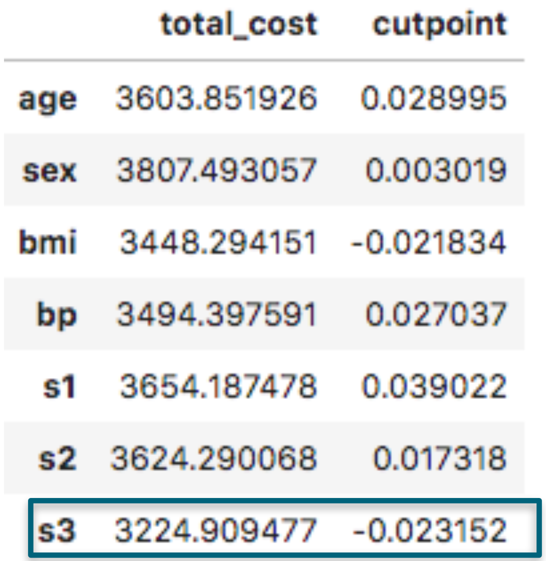
$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} (\hat{y}_{\text{node}} - y^{(i)})^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Root Node



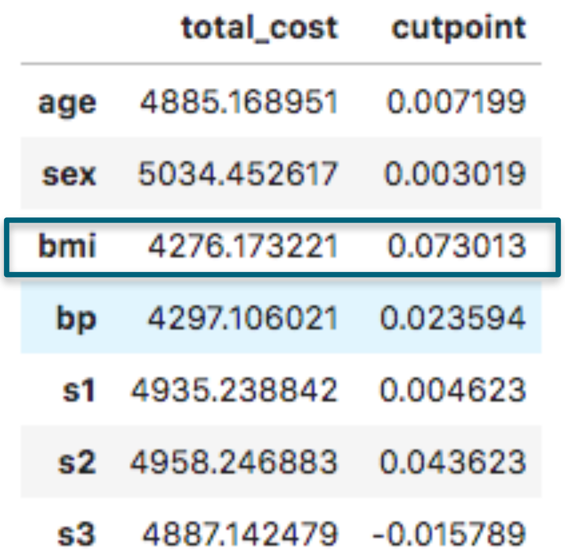
	total_cost	cutpoint
age	5700.035157	0.007199
sex	5918.888900	0.003019
bmi	4279.164764	0.009422
bp	4919.231732	0.023594
s1	5572.695496	0.005999
s2	5658.358682	0.017318
s3	5046.367626	-0.015789

Left Node



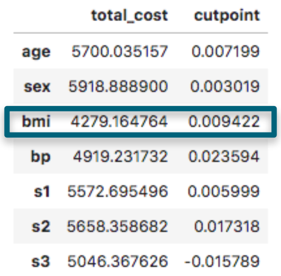
	total_cost	cutpoint
age	3603.851926	0.028995
sex	3807.493057	0.003019
bmi	3448.294151	-0.021834
bp	3494.397591	0.027037
s1	3654.187478	0.039022
s2	3624.290068	0.017318
s3	3224.909477	-0.023152

Right Node

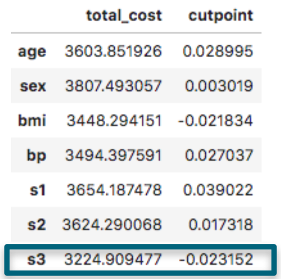


	total_cost	cutpoint
age	4885.168951	0.007199
sex	5034.452617	0.003019
bmi	4276.173221	0.073013
bp	4297.106021	0.023594
s1	4935.238842	0.004623
s2	4958.246883	0.043623
s3	4887.142479	-0.015789

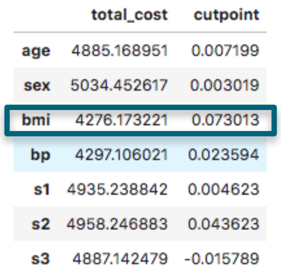
Decision Tree Regressor



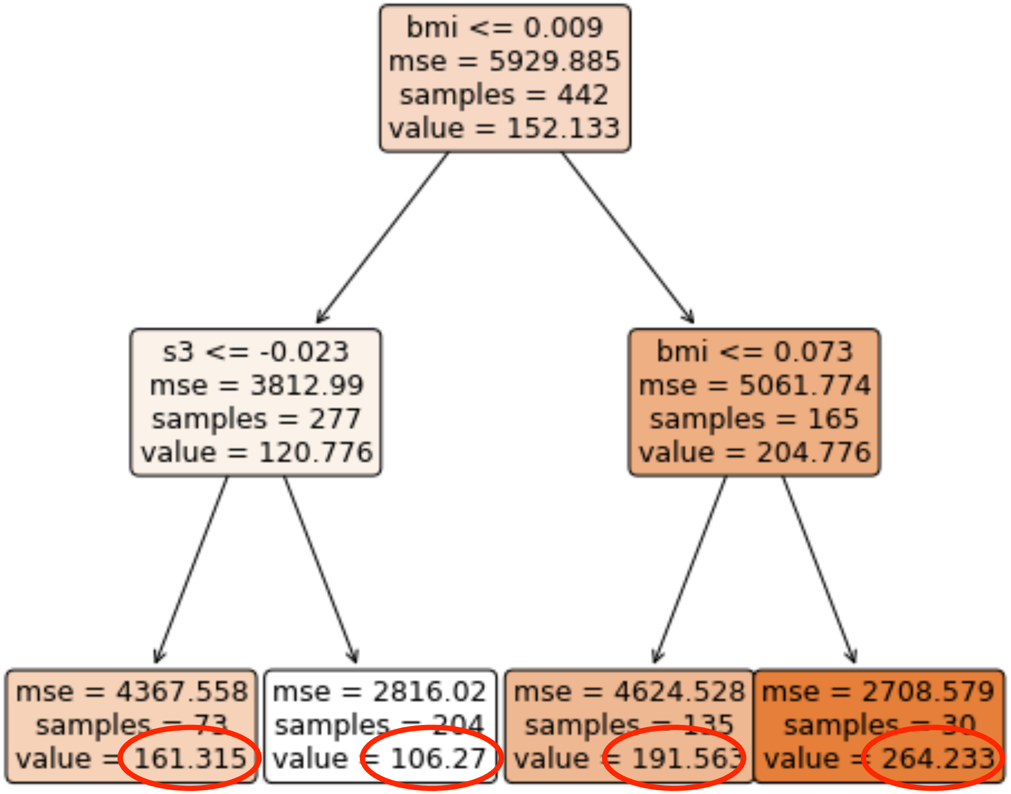
	total_cost	cutpoint
age	5700.035157	0.007199
sex	5918.888900	0.003019
bmi	4279.164764	0.009422
bp	4919.231732	0.023594
s1	5572.695496	0.005999
s2	5658.358682	0.017318
s3	5046.367626	-0.015789



	total_cost	cutpoint
age	3603.851926	0.028995
sex	3807.493057	0.003019
bmi	3448.294151	-0.021834
bp	3494.397591	0.027037
s1	3654.187478	0.039022
s2	3624.290068	0.017318
s3	3224.909477	-0.023152



	total_cost	cutpoint
age	4885.168951	0.007199
sex	5034.452617	0.003019
bmi	4276.173221	0.073013
bp	4297.106021	0.023594
s1	4935.238842	0.004623
s2	4958.246883	0.043623
s3	4887.142479	-0.015789



bmi <= 0.009
mse = 5929.885
samples = 442
value = 152.133

s3 <= -0.023
mse = 3812.99
samples = 277
value = 120.776

bmi <= 0.073
mse = 5061.774
samples = 165
value = 204.776

mse = 4367.558
samples = 73
value = 161.315

mse = 2816.02
samples = 204
value = 106.27

mse = 4624.528
samples = 135
value = 191.563

mse = 2708.579
samples = 30
value = 264.233

Decision Tree Regressor

Classification Tree Regressor

Difference in Cost Function
Regression:

$$J(k,\ t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Classification:
Impurity in Target label

$$J(k,\ t_k) = \frac{n_L}{n_P}\text{Impurity}_L + \frac{n_R}{n_P}\text{Impurity}_R$$

Classification Tree Regressor

Difference in Cost Function
Regression:

$$J(k,\ t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Classification:
Impurity in Target label

$$J(k,\ t_k) = \frac{n_L}{n_P}\text{Impurity}_L + \frac{n_R}{n_P}\text{Impurity}_R$$

$$\begin{aligned} \text{Gini} &= 1 - \sum_{i=1}^{k} p_i^2 \\ \text{Entropy} &= \sum_{i=1}^{k} p_i \log(p_i) \end{aligned}$$

Classification Tree Regressor

$$\Delta \text{Impurity} = \text{Impurity}_P - \left(\frac{n_L}{n_P} \text{Impurity}_L + \frac{n_R}{n_P} \text{Impurity}_R \right)$$

$$\text{Maximize}\left(\Delta \text{Impurity}\right)$$

$$\text{Minimize}\left(\tfrac{n_L}{n_P} \text{Impurity}_L + \tfrac{n_R}{n_P} \text{Impurity}_R \right)$$

Classification Tree Regressor

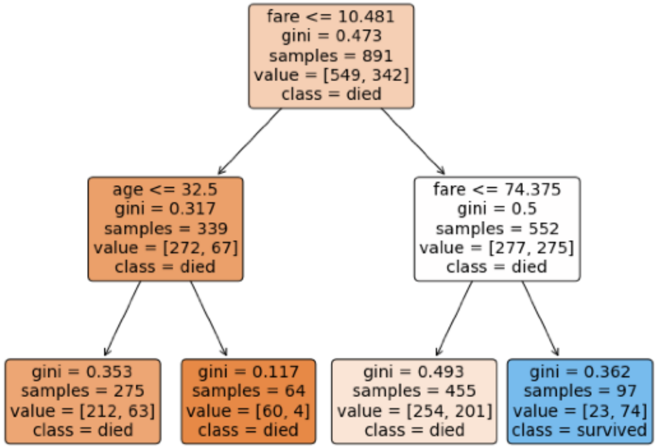
Feature Importance

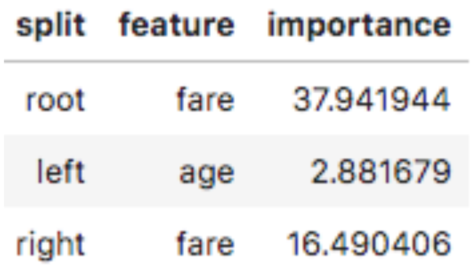
After Feature and Split Point found:

$$\text{Feature Importance} = \frac{n_P}{n}\left(\text{Impurity}_P - \frac{n_L}{n_P}\text{Impurity}_L - \frac{n_R}{n_P}\text{Impurity}_R\right)$$

Where n_p is number of data points in parent node before splitting, and n is number of data points in entire data set.

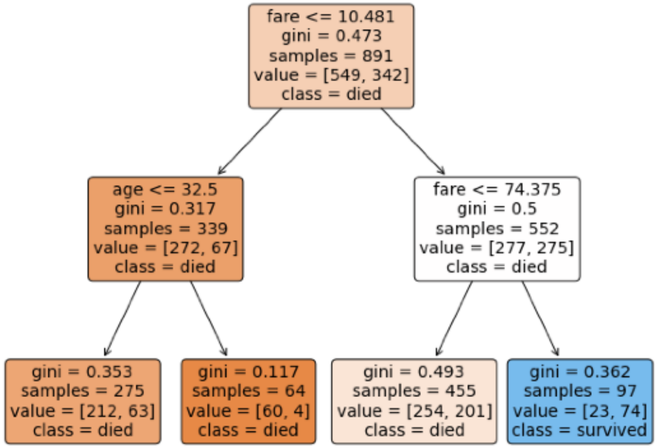
First splits have larger leading weight (=1).
Subsequent splits have smaller weights.

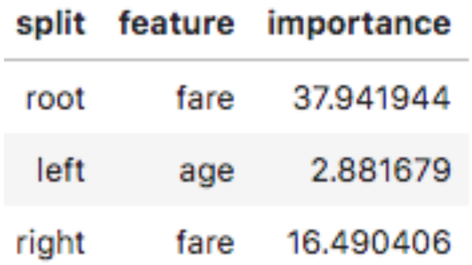




split	feature	importance
root	fare	37.941944
left	age	2.881679
right	fare	16.490406

Feature Importance





split	feature	importance
root	fare	37.941944
left	age	2.881679
right	fare	16.490406

Sum together importance values by feature,

age:	2.881679
fare:	54.432351

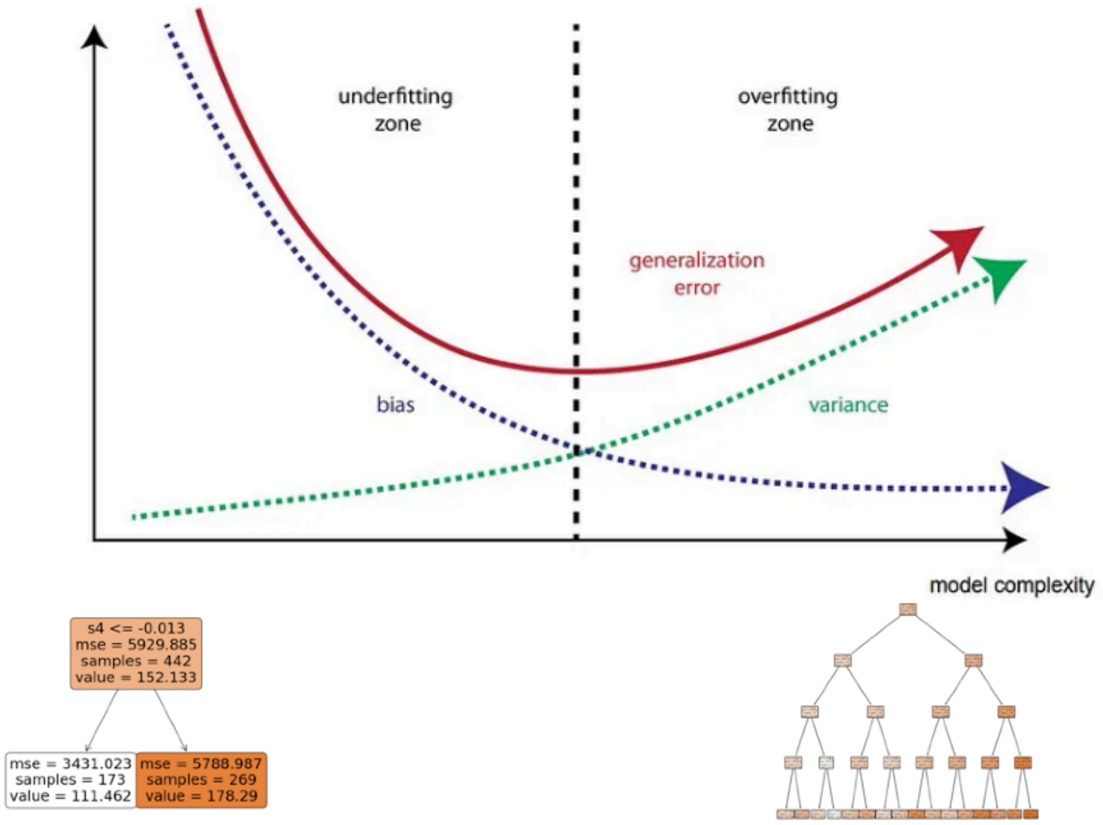
and normalize by sum over all feature.

fare:	0.949721
age:	0.050279

Greedy

At each step of the tree building process, the best split is made at that particular step, rather than looking ahead and selecting a split that will lead to a better tree in some future step.

Complexity


underfitting zone
overfitting zone
generalization error
bias
variance
model complexity
s4 <= -0.013
mse = 5929.885
samples = 442
value = 152.133
mse = 3431.023
samples = 173
value = 111.462
mse = 5788.987
samples = 269
value = 178.29

Implementation

- Will easily overfit without guidance

Implementation

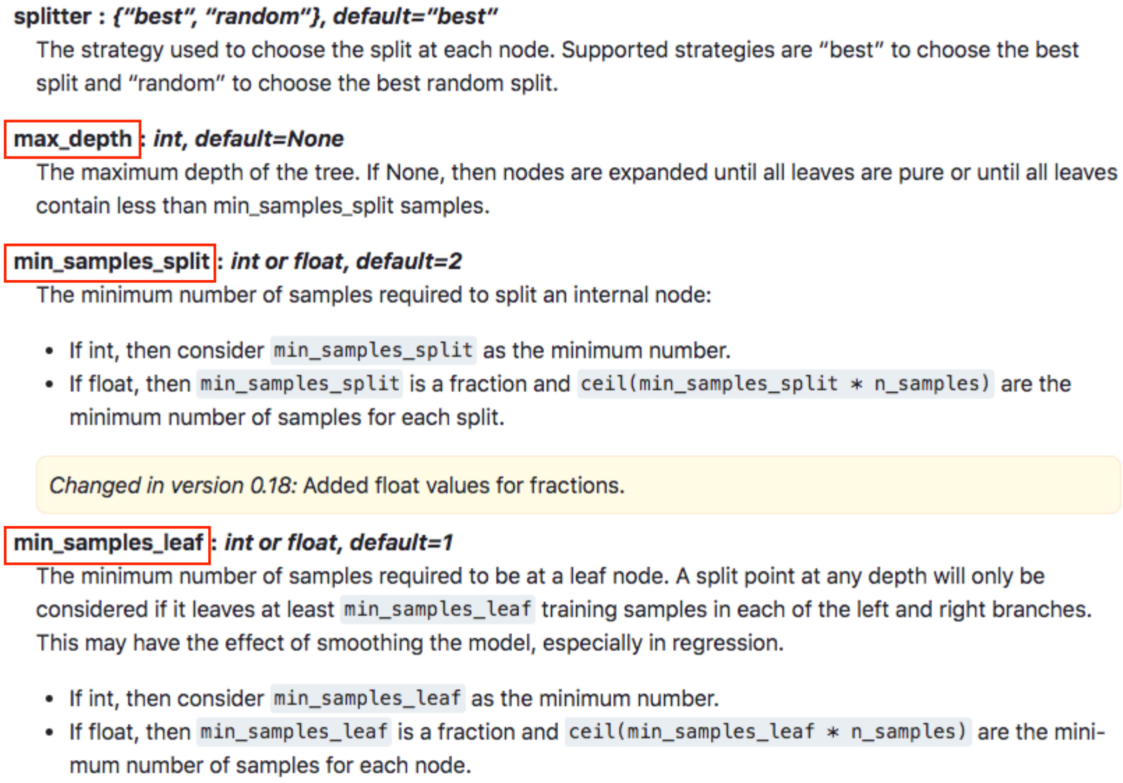
- Will easily overfit without guidance
- How is the optimal, most generalizable, least overfit decision tree found?

Implementation

- Will easily overfit without guidance
- How is the optimal, most generalizable, least overfit decision tree found?
- Hyperparameter tuning
  - Decision Tree Regressor
  - Decision Tree Classifier

Hyperparameter Tuning

- Limit depth…



splitter : *{"best", "random"}, default="best"*

The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

max_depth : *int, default=None*

The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

min_samples_split : *int or float, default=2*

The minimum number of samples required to split an internal node:

- If int, then consider `min_samples_split` as the minimum number.
- If float, then `min_samples_split` is a fraction and `ceil(min_samples_split * n_samples)` are the minimum number of samples for each split.

Changed in version 0.18: Added float values for fractions.

min_samples_leaf : *int or float, default=1*

The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least `min_samples_leaf` training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression.

- If int, then consider `min_samples_leaf` as the minimum number.
- If float, then `min_samples_leaf` is a fraction and `ceil(min_samples_leaf * n_samples)` are the minimum number of samples for each node.