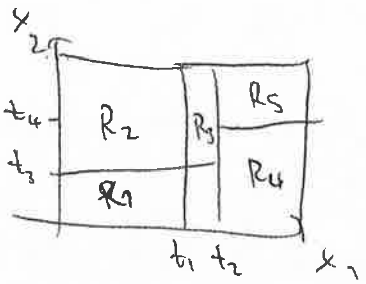
Random Forest

Chapter 10: Boosting and Additive Trees.

Chapter 9(.2) Tree based methods

CART models

Idea is to separate the feature space into several regions, ~~for~~ like below



And then for each region assign a constant value Y_i, that is, for a sample X,

$$Y = \hat{f}(X) = \sum_{m=1}^{5} c_m \underbrace{I}_{\text{indicator function}}\{(X_1, X_2) \in R_m\}.$$

The question now is: how does one grow a regression tree? ~~Given~~ Let $\{(x_i, y_i)\}$ be a dataset with N samples in $\mathbb{R}^p$, $(x_i, y_i) \in \mathbb{R}^{p+1}$

The idea is to use a greedy algorithm in the following manner:

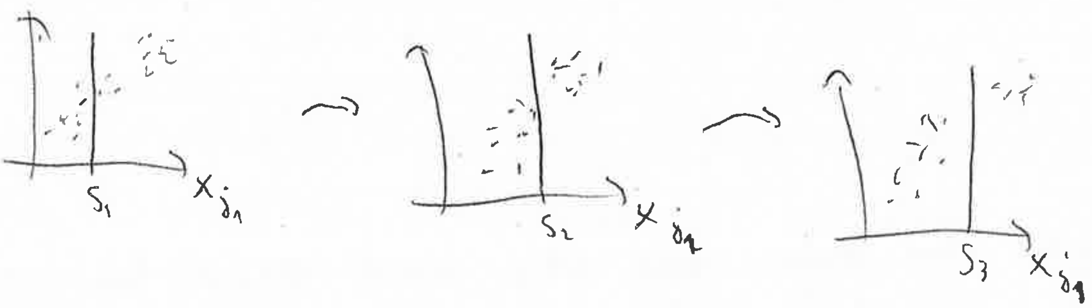
Let j be a splitting variable and s a split point, ~~that~~ means $j \in \{1, \dots, N\}$ and $s \in \mathbb{R}$. Define the pair of half-planes

$$R_1(j,s) = \{X \mid X_j \leq s\} \text{ and } R_2(j,s) = \{X \mid X_j > s\}.$$

②

Then ~~fan~~ we seek the optimal variable j and s such that

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$



For any j,s, the inner minimization is solved by

$$\hat{c}_1=\text{ave}(y_i\mid x_i\in R_1(j,s)) \text{ and } \hat{c}_2=\text{ave}(y_i\mid x_i\in R_2(j,s))$$

For each variable j, the determination of s is done quickly so the determination of the best pair (j,s) is feasible.

+ How large should the tree be?

→ Depends on the data.

The ~~pef~~ preferred strategy is to grow a large tree T_0, stopping the splitting only when some node minimum size is ~~ach~~ reached, i.e., the minimum number of samples in a region for splitting. After that, the tree is pruned using cost-complexity prunning.

Def.: A sub-tree $T\subset T_0$ is any tree that can be obtained by pruning T_0, that is, collapsing any number of its internal (non-terminal) nodes.

Let us index terminal nodes by m, where node m represents region R_m. Let $|T|$ denote the number of terminal nodes in T. Defining

$$N_m = \#\{x_i \in R_m\} \qquad \leftarrow \text{cardinality of } R_m$$

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i \qquad \leftarrow \text{average of points in } R_m$$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2 \qquad \leftarrow \text{mean squared error}$$

Then the cost complexity criterion is

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|$$

The idea is to find for each α a tree $T_\alpha \subset T_0$ that minimizes $C_\alpha(T)$. The bigger α is, the smaller T_α should be. And so conversely. By construction, when $\alpha = 0$, the optimal tree is the full tree.

* How to choose α?

For each α one can show that there exists the optimal tree T_α (probably because of convexity). Then to find T_α we use <u>weakest link pruning</u>. We produce a sequence of trees by collapsing the internal node that produces the smallest per-node increase in $\sum_m N_m Q_m(T)$, until we produce the single-node (root) tree. One can then show that the optimal tree is contained in this sequence. In practice, the parameter α is estimated doing a cross-validation with sum of squares.

Classification trees

The difference now is in the impurity measure $Q_m(T)$. Let m be a node representing region R_m with N_m observations, let

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k),$$

the proportion of class k in ~~node~~ node m. The observation in node m is classified as the majority class in R_m, i.e., class $k(m) = \arg\max_k \hat{p}_{mk}$.

Different measures of node impurity are used:

Misclassification error: $\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{m k(m)}$

Gini index: $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$

Cross-entropy or deviance: $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

For guiding the cost-complexity pruning the misclassification rate is used.

Chapter 15: Random Forests

Algorith 15.1 Let B be the number of trees

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample Z^* of size N from the training data

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_{b=1}^{B}$.

And then, to make a prediction at a new point x:

Regression: $\hat{f}_{rf}^{B}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{rf}^{B}(x) = \text{majority vote } \{\hat{C}_b(x)\}_{b=1}^{B}$.

Variable importance (15.3.2)

. It's calculated in the same way as in the gradient boosting models and will be discussed later.

Another type of variable importance measure uses the OOB (out-of-bag) samples, to measure the prediction strength of each variable. After the tree is fit, within the OOB samples, the j-th variable is ~~perm~~ randomly permuted and the accuracy is computed. The decrease in accuracy due to the permutation is ~~given b~~ averaged across all trees and is then used as a measure of importance of variable j in the random forest.

Sklearn feature importance

Feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node. And the node probability can be calculated by the number of samples that reach the node, divided by the total number of samples.

The impurity can be calculated in the following manners:

Impurity	Task	Formula	Description
Gini Impurity	Classification	$\sum_{i=1}^{C} f_i(1-f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels
Entropy	Classification	$\sum_{i=1}^{C} -f_i \log(f_i)$	"
MSE	Regression	$\frac{1}{N}\sum_{i=1}^{N}(y_i-\mu)^2$	y_i is label for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$
MAE	Regression	$\frac{1}{N}\sum_{i=1}^{N}\lvert y_i-\mu\rvert$	"

Implementation in sklearn

In this step we divide in the importance of a node in a tree, importance of a feature in each tree and the total importance.

(i) For a single node the importance feature is calculated in the following manner using Gini Importance (classification task).

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)},$$

where:

ni_j = the importance of node j
w_j = weighted number of samples in node j
C_j = impurity in node j (error reduction)
left and right = respective values in the children nodes.

Then for feature j, its importance is calculated by

$$fi_j = \frac{\sum_{k:\text{ node } k \text{ splits on feature } j} ni_k}{\sum_{k \in \text{all nodes}} ni_k},$$

where:

fi_j = the importance of feature j
ni_k = the importance of node j.

These are then normalized so all the importance features are between 0 and 1. And finally, as a final output, the importance of features are averaged across all trees.

Remark: When calculating the node importance, the samples that are used to calculate the ~~node import~~ impurity are those which are routed to the ~~numb~~ node. The weighted number of samples in node j ~~are~~ is just the ratio of number of samples routed to that node by the total number of samples.

Example

Let us consider a very simple example of a decision tree. We will apply the CART algorithm to find the splitting variables. Consider the following dataset with 2 features and a continuous prediction value:

	X_1	X_2	Y
x_1	1	4	3
x_2	3	0	6
x_3	5	2	10

Table 1

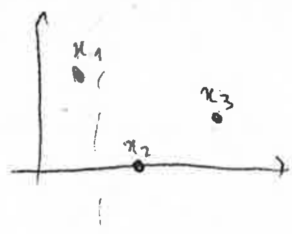


Figure 1

We will build only one ~~three~~ tree to its maximal depth. Let us consider the first splitting. We have to find the best parameters j and s such that

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right].$$

It can be shown that

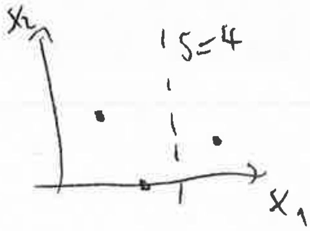
$$c_i = \text{ave}(y_i \mid x_i\in R_i(j,s)).$$

Thus, for the first splitting we can consider the following values

j	s
1	2
1	4
2	1
2	3

We select ~~thes~~ the values for s by inspecting the graphic in figure 1. Therefore, for $j=1$, we have two possible splittings:

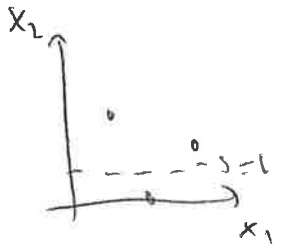
and



Calling $err(j,s)=\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2$, we have

- $err(1,2) = \overset{0}{\cancel{(3-3)}}{}^2+(6-8)^2+(10-8)^2 = 8$, $c_1=3$, $c_2=8$ and $R_1(1,2)=\{(1,4)\}$, $R_2(1,2)=\{(3,0),(5,2)\}$.
- $err(1,4) = (3-4.5)^2+(6-4.5)^2+(10-10)^2 = 1.5^2+1.5^2 = 5.5$, $c_1=4.5$, $c_2=10$, $R_1(1,4)=\{(1,4),(3,0)\}$, $R_2=\{(5,2)\}$.

And lastly, for $j=2$



and

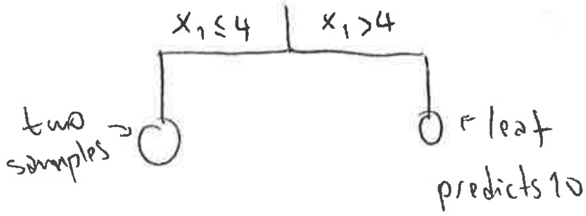
- $err(2,1) = (6-6)^2 + (3-7.5)^2 + (10-7.5)^2 = 4.5^2 + 2.5^2 = 26.5$

  $c_1 = 6,\ c_2 = 4.5,\ R_1(2,1) = \{(3,0)\},\ R_2(2,1) = \{(1,4),(5,2)\}$

- $err(2,3) = (6-8)^2 + (10-8)^2 + (3-3)^2 = 4+4 = 8$

  $c_1 = 8,\ c_2 = 3,\ R_1(2,3) = \{(3,0),(5,2)\},\ R_2(2,3) = \{(1,4)\}.$

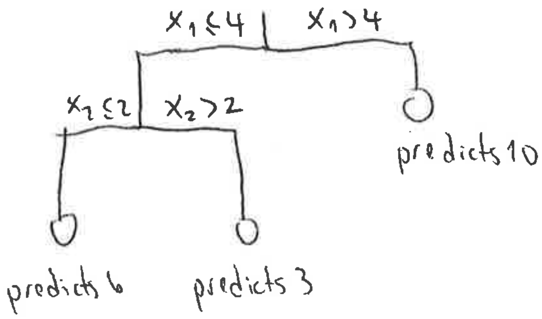
Thus, among all possibilities, the best pair that gives the minimal error is $(1,4)$. So then we have the first splitting:

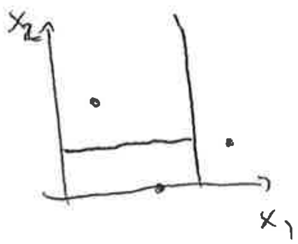


And only the following samples are left on the left node:

x_1	x_2	y
3	0	6
1	4	3

We can pick a splitting among $(1,2)$ or $(2,2)$, since the error will be 0. Selecting $(2,2)$, we get the following tree.





The values tha are predicted in each leaf are the mean of the samples in the leaf, but since we have only one value, the mean is the sample value. And calculating the feature importances, we get:

For node 1, representing variable X_1

$$ni_1 = w_1 C_1 - w_{left(1)} C_{left(1)} - w_{right(1)} C_{right(1)}$$

$w_1 = 1$ $\quad w_{left(1)} = \frac{1}{3}$ $\quad C_{left(1)} = 0$

$C_1 = \frac{1}{3} \sum_{i=1}^{3} (y_i - 9.5)^2 = 18.25$ $\quad w_{right(1)} = \frac{2}{3}$ $\quad C_{left(2)} = \frac{1}{2} \sum_{i=1}^{2} (y_i - 4.5)^2 = 2.25$

$$\Rightarrow ni_1 = 18.25 - \frac{2}{3} 2.25 = 16.75$$

For node 2, representing variable X_2

$$ni_2 = w_2 C_2 - w_{left(2)} C_{left(2)} - w_{right(2)} C_{right(2)}$$

$w_2 = \frac{2}{3}$ $\quad w_{left(2)} = \frac{1}{3}$ $\quad C_{left(2)} = 0$

$C_2 = 2.25$ $\quad w_{right(2)} = \frac{1}{3}$ $\quad C_{right(2)} = 0$

$$\Rightarrow ni_2 = \frac{2}{3} \cdot (2.25) = 1.5$$

And the importance for each feature is given by

$$fi_1 = \frac{16.75}{(1.5 + 16.75)} = 0{,}9178 \qquad fi_2 = 1 - fi_1 = 0{,}0822$$

In that way, variable X_1 has an importance ~~feature~~ of ~~0.457~~ 0.9178 and variable X_2 has an importance of 0.0822