Московский государственный университет им. М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Решающие деревья

(CART алгоритм)

Риве Алина, 316

2021

Содержание

Решающие деревья.......................................................................................3

CART алгоритм............................................................................................5

Реультаты......................................................................................................7

Литература...................................................................................................13

Листинг........................................................................................................14

**Решающие деревья**

Решающее дерево (Decision tree) — решение задачи обучения с учителем, основанный на том, как решает задачи прогнозирования человек. В общем случае — это k-ичное дерево с решающими правилами в нелистовых вершинах (узлах) и некотором заключении о целевой функции в листовых вершинах (прогнозом). Решающее правило — некоторая функция от объекта, позволяющее определить, в какую из дочерних вершин нужно поместить рассматриваемый объект. В листовых вершинах могут находиться разные объекты: класс, который нужно присвоить попавшему туда объекту (в задаче классификации), вероятности классов (в задаче классификации), непосредственно значение целевой функции (задача регрессии).

Деревья классификации и регрессии являются одним из наиболее популярных методов решения многих практических задач, что обусловлено следующими причинами:

1. Деревья решений позволяют получать очень легко интерпретируемые модели, представляющие собой набор правил вида “если…, то…”. Интерпретация облегчается, в том числе, за счет возможности представить эти правила в виде наглядной древовидной структуры.
2. В силу своего устройства деревья решений позволяют работать с переменными любого типа без необходимости какой-либо предварительной подготовки этих переменных для ввода в модель (например, логарифмирование, преобразование категориальных переменных в индикаторные, и т.п.).
3. Исследователю нет необходимости в явном виде задавать форму взаимосвязи между откликом и предикторами, как это, например, происходит в случае с обычными регрессионными моделями. Это оказывается особенно полезным при работе с данными большого объема, о свойствах которых мало что известно.
4. Деревья решений, по сути, автоматически выполняют отбор информативных предикторов и учитывают возможные взаимодействия между ними. Это, в частности, делает деревья решений полезным инструментом разведочного анализа данных.
5. Деревья решений можно эффективно применять к данным с пропущенными значениями, что очень полезно при решении практических задач, где наличие пропущенных значений – это, скорее, правило, чем исключение.
6. Деревья решений одинаково хорошо применимы как к количественным, так и к качественным зависимым переменным.

К недостаткам этого класса моделей иногда относят нестабильность и невысокую точность предсказаний, что, как будет показано ниже, не всегда, но часто подтверждается. По своей сути, деревья используют “наивный подход” (naive approach) в том смысле, что они исходят из предположения о взаимной независимости признаков. Поэтому модели регрессионных деревьев статистически наиболее работоспособны, когда комплекс анализируемых переменных является не слишком мультиколлинеарным или имеется регулярная внутренняя множественная альтернатива в исходной комбинации признаков.

Деревья решений бывают двух основных типов:

* Дерево для классификации, когда предсказываемый результат является классом, к которому принадлежат данные;
* Дерево для регрессии, когда предсказываемый результат можно рассматривать как вещественное число.

Некоторые методы позволяют построить более одного дерева решений (ансамбли деревьев решений):

* Бэггинг над деревьями решений, наиболее ранний подход. Строит несколько деревьев решений, неоднократно интерполируя данные с заменой (бутстреп), и в качестве консенсусного ответа выдаёт результат голосования деревьев (их средний прогноз);
* Классификатор «Случайный лес» основан на бэггинге, однако в дополнение к нему случайным образом выбирает подмножество признаков в каждом узле, с целью сделать деревья более независимыми;
* Бустинг над деревьями может быть использован для задач как регрессии, так и классификации. Одна из реализаций бустинга над деревьями, алгоритм XGBoost, неоднократно использовался победителями соревнований по анализу данных.
* «Вращение леса» — деревья, в которых каждое дерево решений анализируют первым применением метода главных компонент (PCA) на случайные подмножества входных функций.

Есть различные способы выбирать очередной признак:

* Алгоритм ID3, где выбор признака происходит на основании прироста информации (англ. Gain), либо на основании критерия Джини.
* Алгоритм C4.5 (улучшенная версия ID3), где выбор признака происходит на основании нормализованного прироста информации (англ. Gain Ratio).
* Алгоритм CART и его модификации — IndCART, DB-CART.
* Автоматический детектор взаимодействия Хи-квадрат (CHAID). Выполняет многоуровневое разделение при расчёте классификации деревьев;[6]
* MARS: расширяет деревья решений для улучшения обработки цифровых данных.

На практике, в результате работы этих алгоритмов часто получаются слишком подробные деревья, которые при их дальнейшем применении дают много ошибок. Это связано с явлением переобучения. Для сокращения деревьев используют отсечение ветвей.

**CART алгоритм**

Популярный алгоритм построения деревьев решений, который может работать как с дискретной, так и с непрерывной выходной переменной, т.е. решать задачи и классификации, и регрессии.

Он разработан в 1974—1984 годах четырьмя профессорами статистики: Лео Брейманом, Джеромом Фридманом, Чарлзом Стоуном и Ричардом Олшеном.

На сегодняшний день существует большое число алгоритмов, реализующих деревья решений: CART, C4.5, CHAID, CN2 , NewId, ITrule и другие.

Алгоритм строит бинарные деревья решений, которые содержат только два потомка в каждом узле. В процессе работы происходит рекурсивное разбиение примеров обучающего множества на подмножества, записи в которых имеют одинаковые значения целевой переменной.

Особенности алгоритма CART:

* функция оценки качества разбиения;
* механизм отсечения дерева;
* алгоритм обработки пропущенных значений;
* построение деревьев регрессии.

Алгоритм реализует обучение с учителем и использует в качестве критерия для выбора разбиений в узлах индекс чистоты Джини (Gini impurity index).

где T – набор данных, n – количество классов, информация о которых содержится в T, - вероятность (относительная частота) класса i в T.

Если набор Т разбивается на две части Т1 и Т2 с числом примеров в каждом N1 и N2 соответственно, тогда показатель качества разбиения будет равен:

В процессе роста дерева алгоритм CART проводит для каждого узла полный перебор всех атрибутов, на основе которых может быть построено разбиение. Наилучшим считается то разбиение, для которого Ginisplit(T) минимально.

Обозначим N — число примеров в узле — предке, L, R — число примеров соответственно в левом и правом потомке, li и ri — число экземпляров i-го класса в левом/правом потомке. Тогда качество разбиения оценивается по следующей формуле:

Основная идея алгоритма заключается в том, чтобы выбрать такое разбиение из всех возможных в данном узле, чтобы полученные дочерние узлы были максимально однородными. При этом каждое разбиение производится только по одному атрибуту.

Механизм отсечения дерева, оригинальное название minimal cost-complexity tree pruning, – наиболее серьезное отличие алгоритма CART от других алгоритмов построения дерева. CART рассматривает отсечение как получение компромисса между двумя проблемами: получение дерева оптимального размера и получение точной оценки вероятности ошибочной классификации.

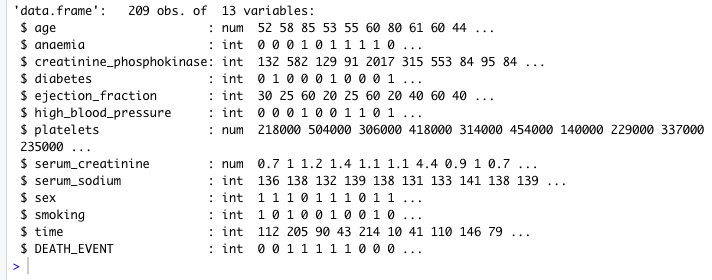
Основная проблема отсечения – большое количество всех возможных отсеченных поддеревьев для одного дерева. Базовая идея метода – не рассматривать все возможные поддеревья, ограничившись только "лучшими представителями".

Перекрёстная проверка – путь выбора окончательного дерева, при условии, что набор данных имеет небольшой объем или же записи набора данных несколько специфические, что разделить набор на обучающую и тестовую выборки невозможно. Этот метод применяется тогда, когда каждая запись в тестовой выборке уникальна или мала.

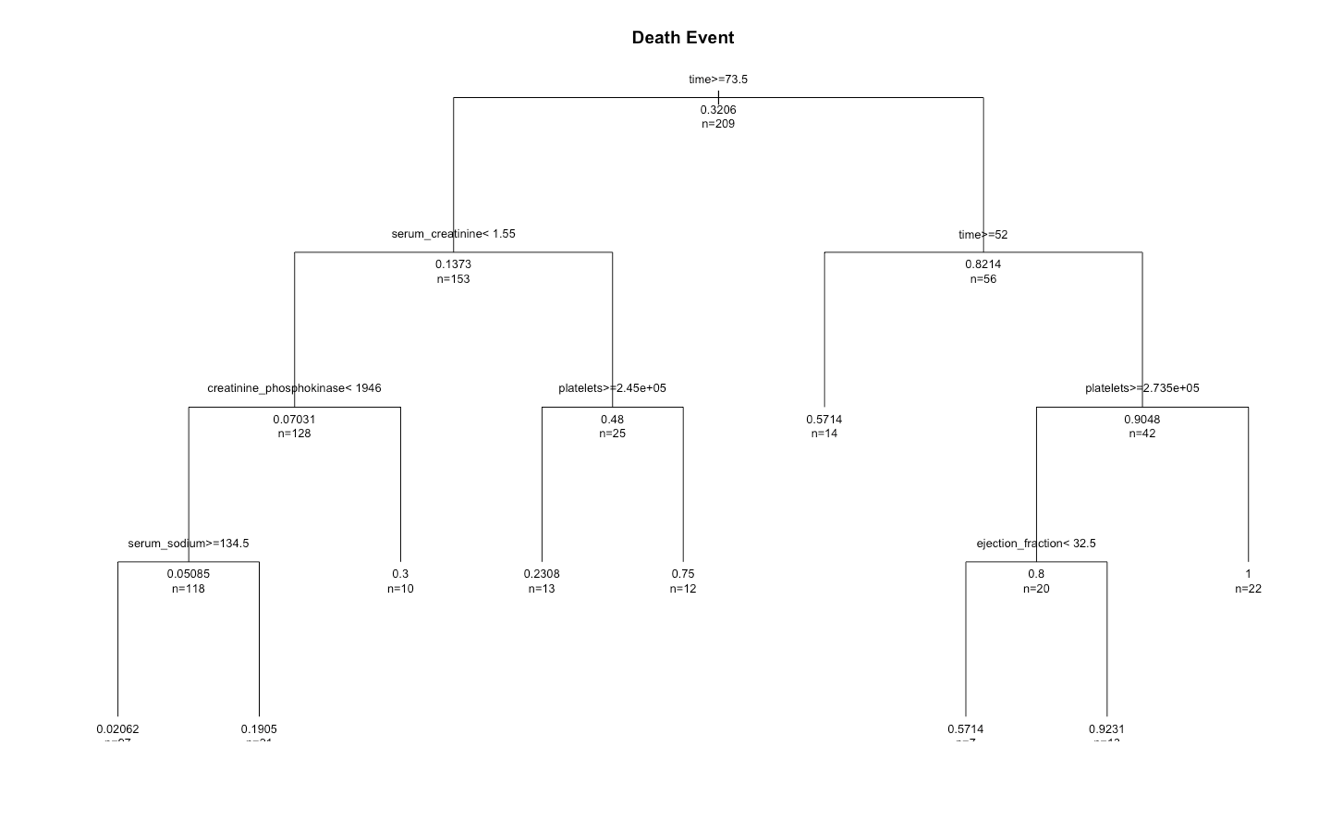
**Результаты**

Решающее дерево построено по CART алгоритму на основе данных о симптомах больных, переживших сердечный приступ. Дерево должно определять, выживет ли человек.

В наборе данных 13 атрибутов. Один из них – DEATH EVENT – событие смерти – целевая функция.

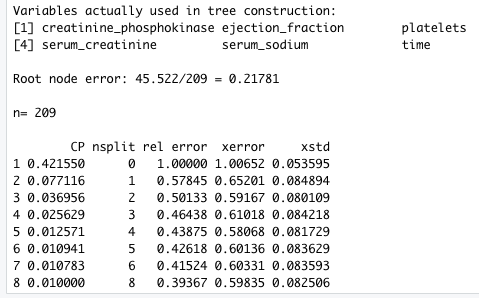


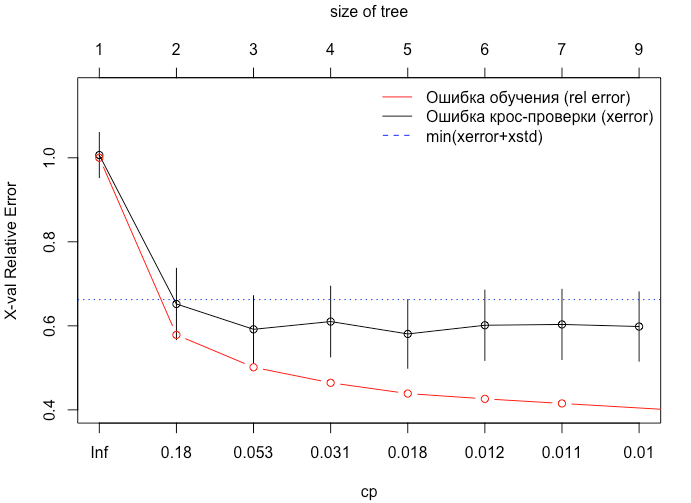
Применяю алгоритм построения дерева по CART алгоритму. В результате получилось, что при его построении использовались только 6 атрибутов, что объясняется тем, как алгоритм выбирает решающее правило по индексу Джини.



Получили дерево из 8 вершин.

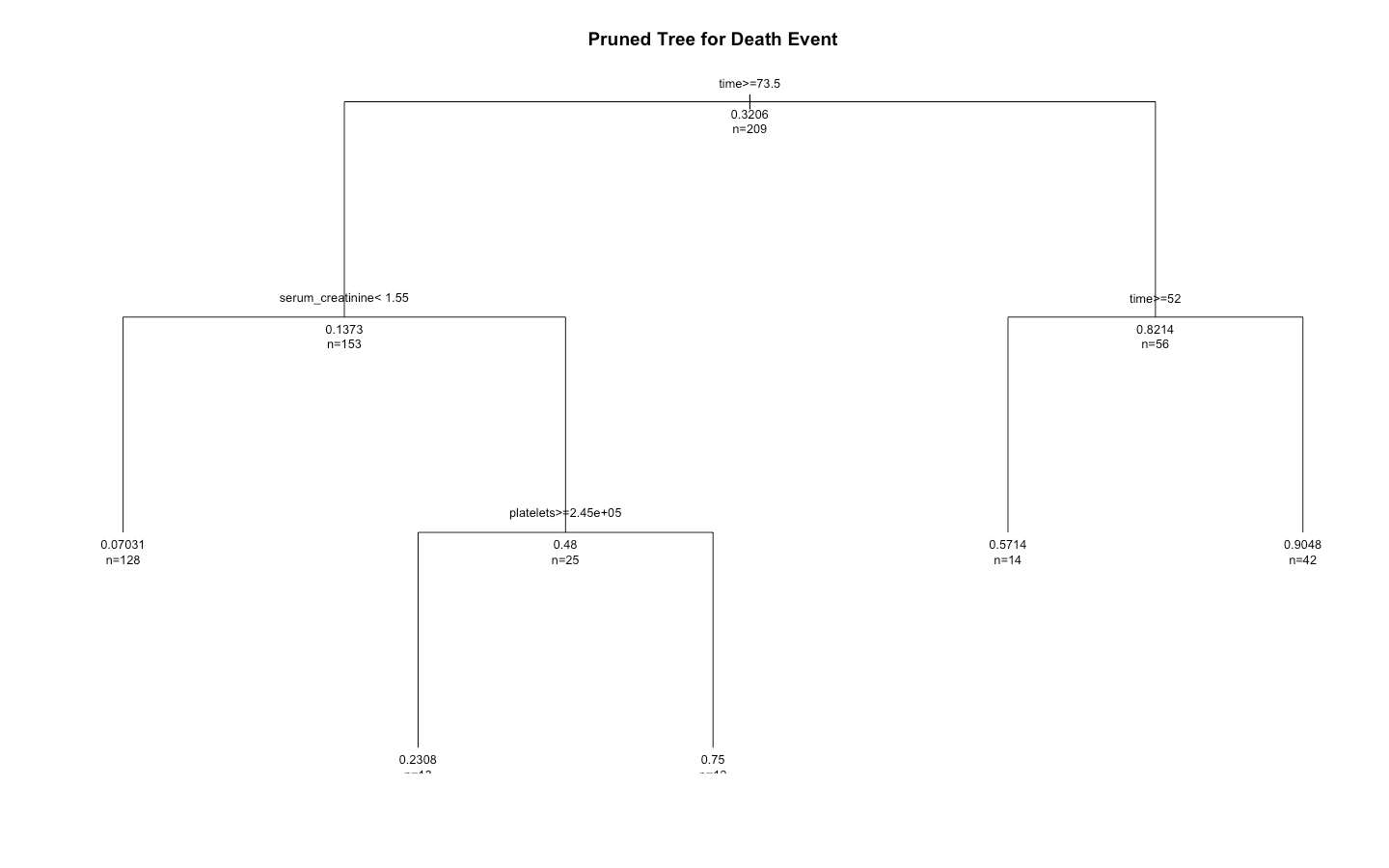
После построения дерева, нас больше всего интересуют значения xerror – ошибка перекрестной проверки; xstd – стандартное отклонение ошибки перекрестной проверки, rel error – ошибка обучения, cp – относительный параметр стоимости сложности (чем больше cp, тем меньше nsplit – размер разделения дерева). Именно по значению ошибки перекрестной проверки будем определять cp, по которому будет производится отсечение.

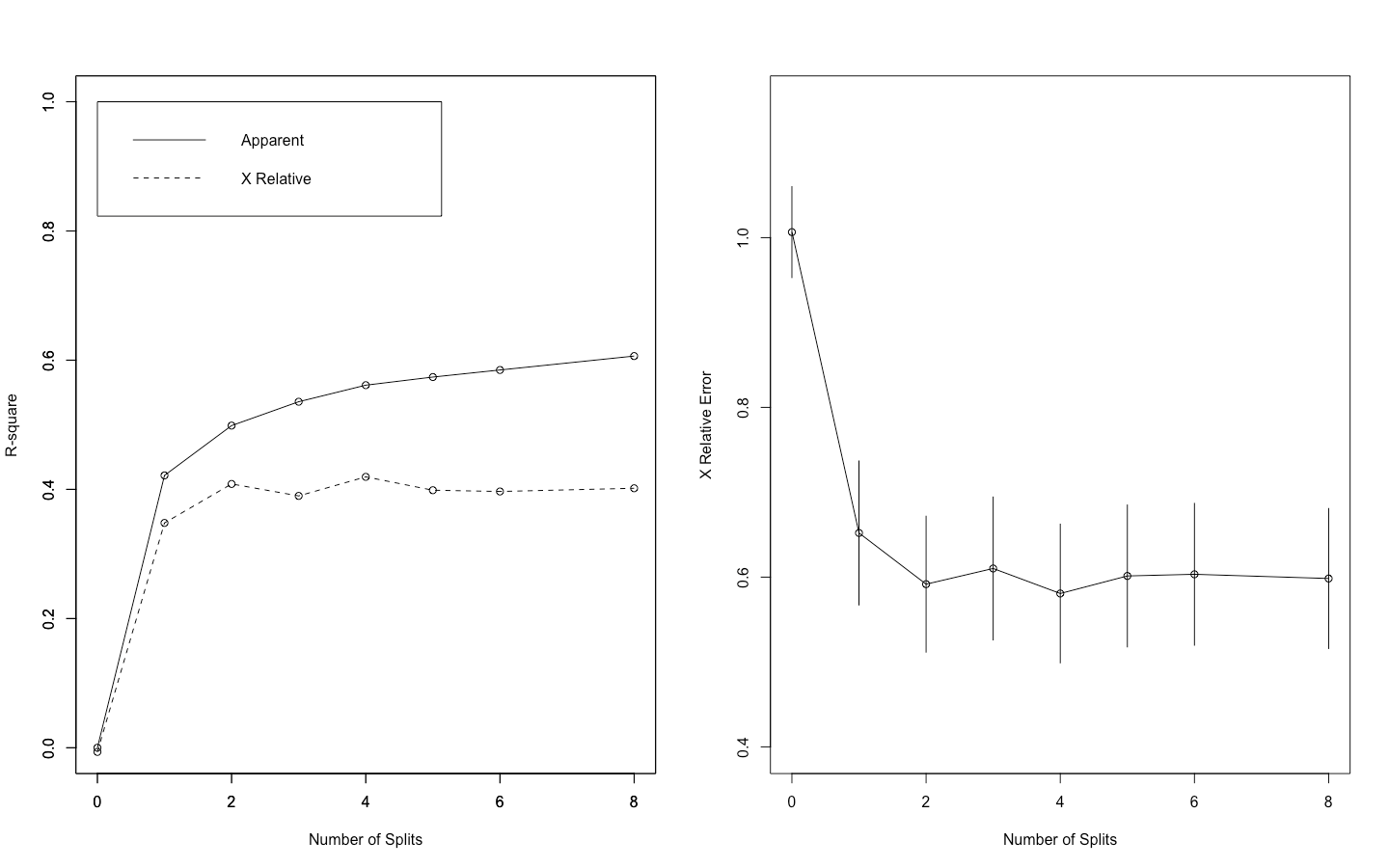




Из таблицы и графика видно, что минимальная ошибка перекрестной проверки достигается при nsplit = 4, cp = 0.012571.

Строим дерево без той части, которая должна быть отсечена.

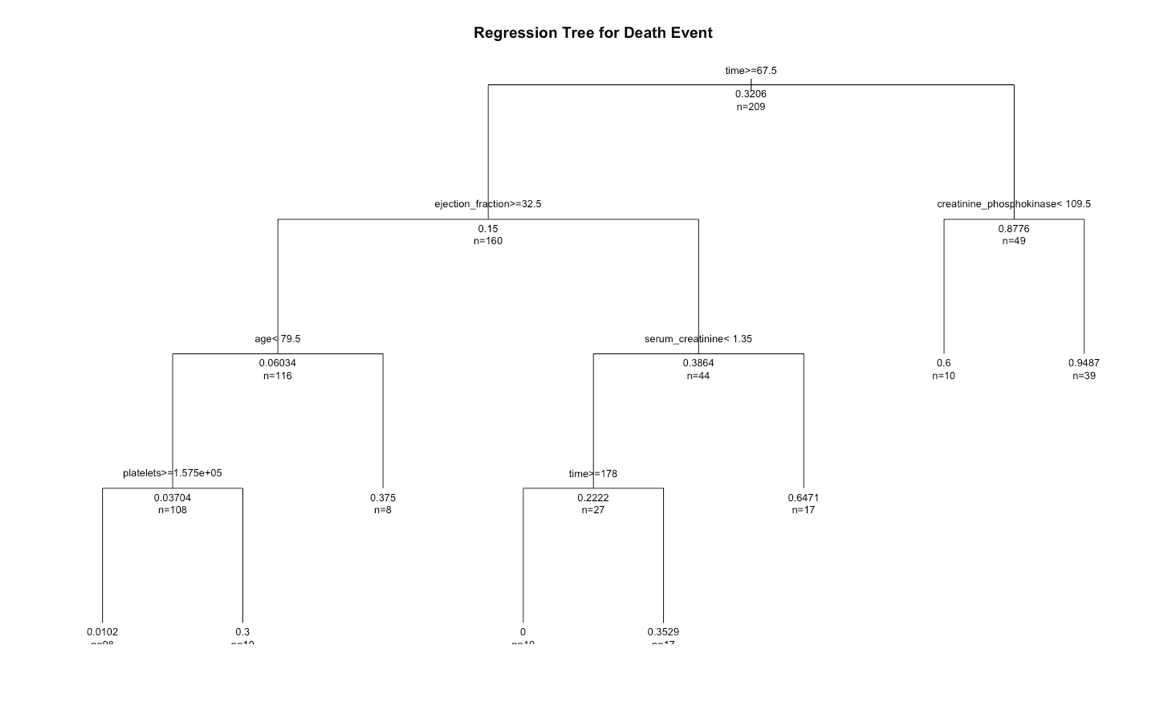


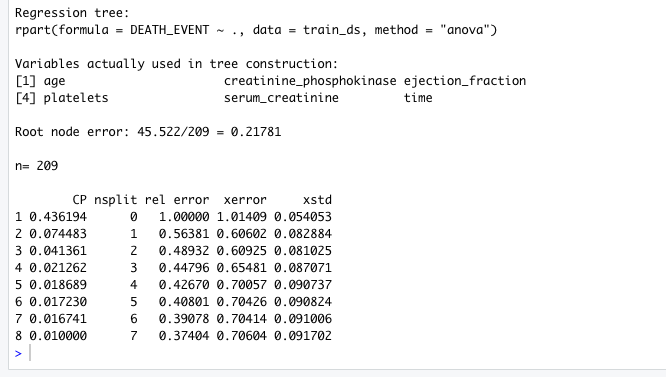


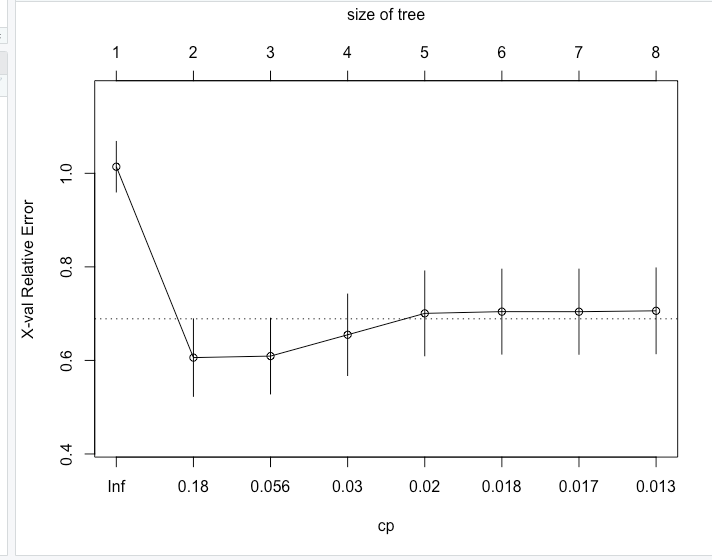
Рассмотрим графики для коэффициента детерминации R-squared – в этом случае он показывает как хорошо модель подходит к данным. Очевидно, что хотя r-squared растет с увеличением разбиений, его значение около 0,6, что говорит о приемлемом соответствии модели данным, но не о хорошем.

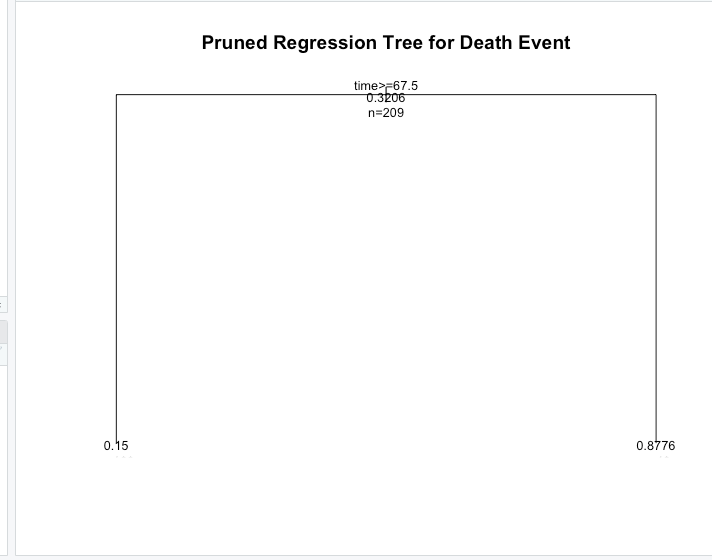
Второй график – аналог одного из уже рассмотренных, но здесь по оси OX не cp, а количество разбиений.

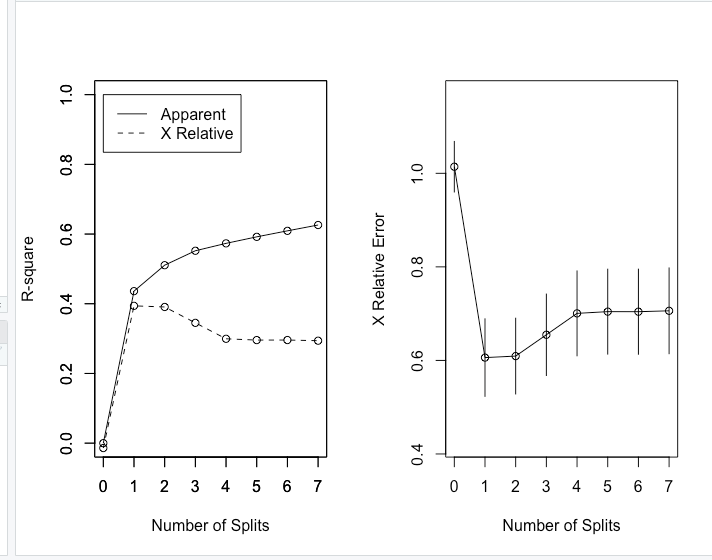
Для того, чтобы подтвердить, что дерево решений не всегда стабильная модель, покажу анаолгичные результаты для того же набора данных, полученные при другом запуске алгоритма построения дерева.











Возможно, если преобразовать тип DEATH\_EVENT, модель будет лучше соответствовать данным. Но так как одним из преимуществ решающих деревьев является их способность работать с любыми типами без преобразования, было интересно применить алгоритм именно к не преобразованным данным.

Литература

1. <http://datascientist.one/cart-algorithm/>
2. <https://basegroup.ru/community/articles/math-cart-part1>
3. <https://basegroup.ru/community/articles/math-cart-part2>
4. <https://ru.wikipedia.org/wiki/CART_(алгоритм)>

Листинг