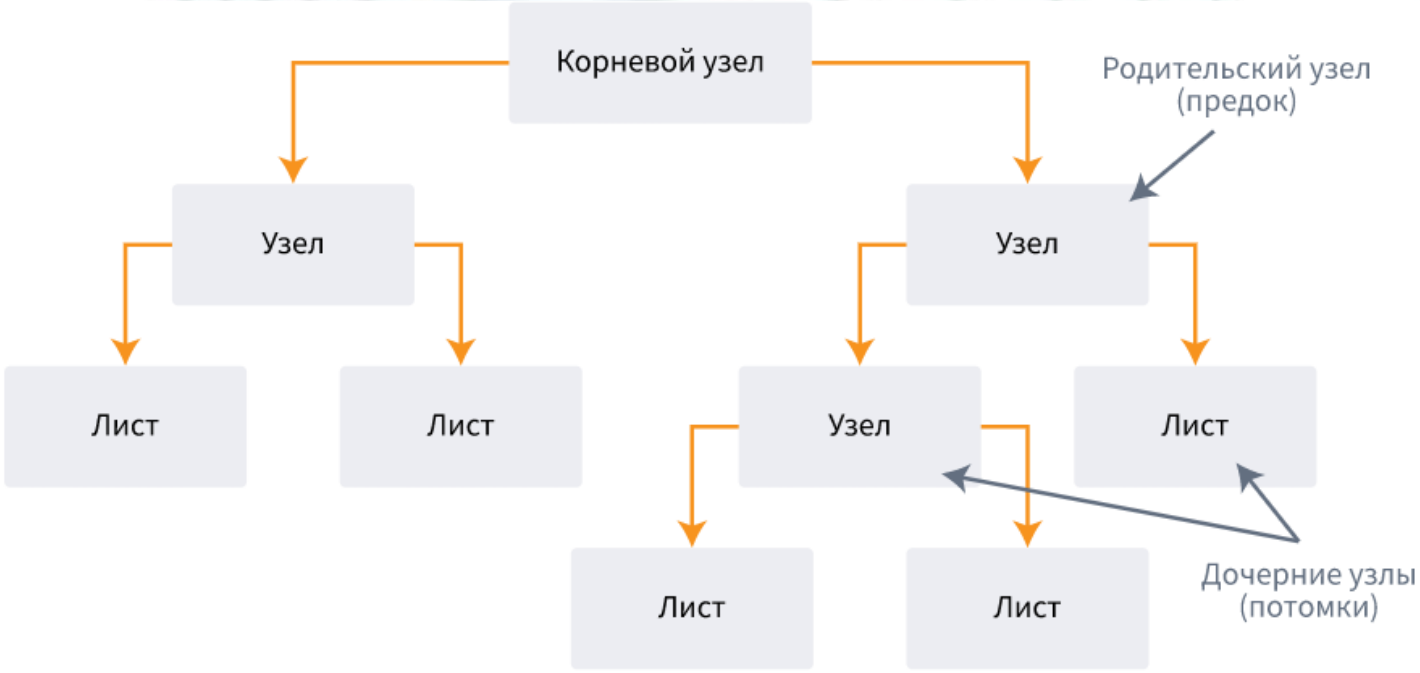
Основные понятия и определения деревьев решений

ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

Введение в деревья решений

Деревья решений (decision trees) относятся к числу самых популярных и мощных инструментов Data Mining, позволяющих эффективно решать задачи классификации и регрессии. В отличие от методов, использующих статистический подход, таких как классификатор Байеса, линейная и логистическая регрессия, деревья решений основаны на **машинном обучении** и в большинстве случаев не требуют предположений о статистическом распределении значений признаков. В основе деревьев решений лежат решающие правила вида «если… то…», которые могут быть сформулированы на естественном языке. Поэтому деревья решений являются наиболее наглядными и легко интерпретируемыми моделями



Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — атрибуты, по которым различаются случаи

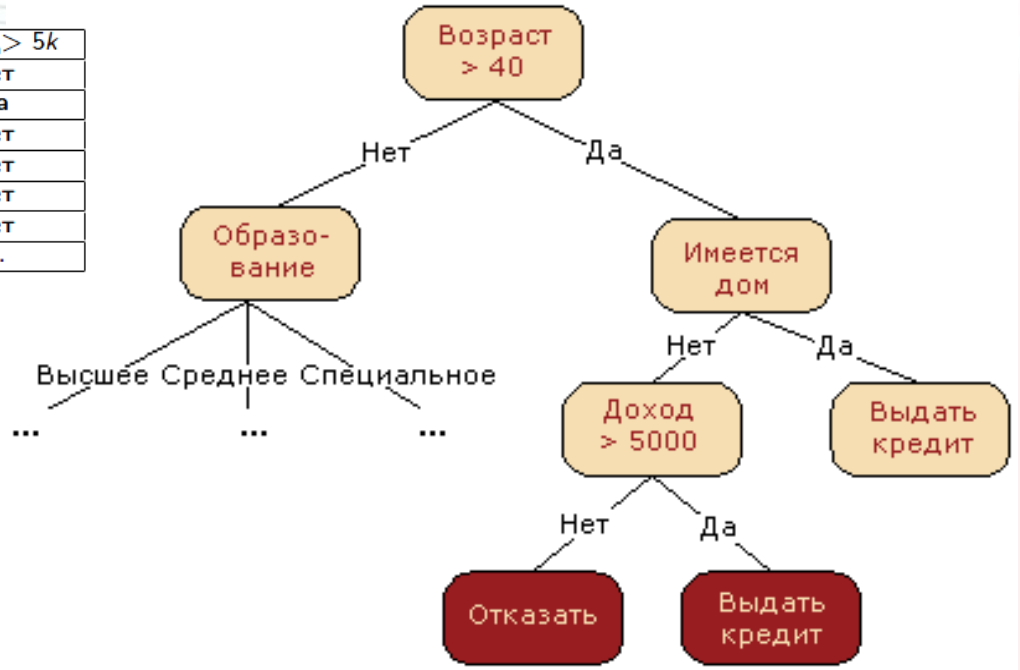
Введение в деревья решений

Дерево принятия решений (также может называться деревом классификации или регрессионным деревом) — средство поддержки принятия решений, использующееся в машинном обучении.
Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.
Подобные деревья решений широко используются в интеллектуальном анализе данных.
Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе.
Метод построения деревьев решений является нисходящим: мы начинаем с исходного входного пространства X и разделяем его на две дочерние области с помощью порогового значения одного признаков. Затем мы берем один из этих дочерних регионов и можем разделить через новый порог. Мы продолжаем обучение нашей модели рекурсивным способом, всегда выбирая конечный узел, элемент и порог для формирования нового разделения

Введение в деревья решений

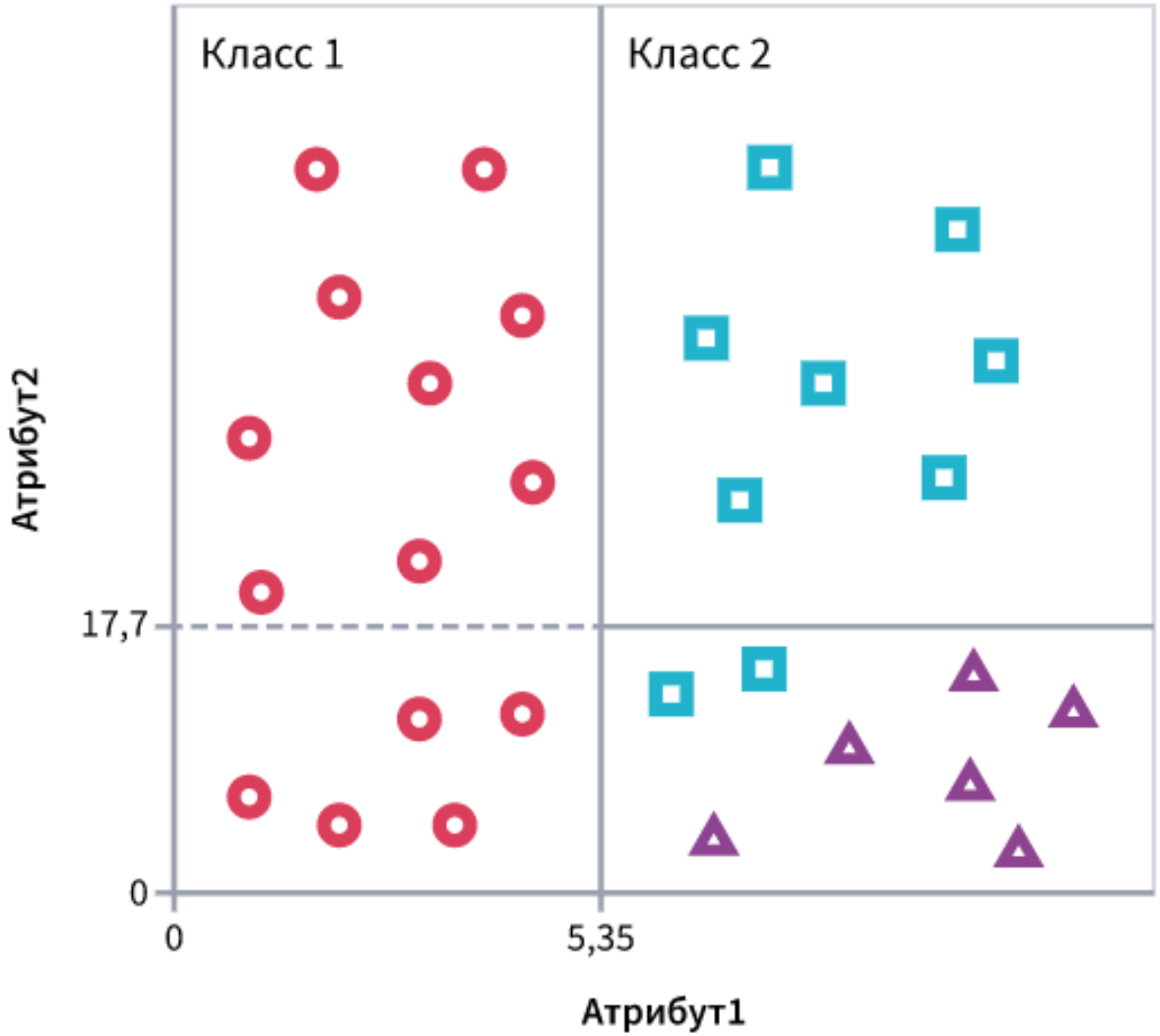
Строятся деревья решений на основе обучения с учителем.
Структурно дерево решений состоит из объектов двух типов — узлов (node) и листьев (leaf). В узлах расположены решающие правила и подмножества наблюдений, которые им удовлетворяют. В листьях содержатся классифицированные деревом наблюдения: каждый лист ассоциируется с одним из классов, и объекту, который распределяется в лист, присваивается соответствующая метка класса.

Решение	Дом	Возраст > 40	Образование	Доход > 5k
Отказать	Нет	Да	Среднее	Нет
Выдать	Нет	Да	Специальное	Да
Выдать	Да	Да	Среднее	Нет
Выдать	Да	Нет	Высшее	Нет
Отказать	Да	Нет	Среднее	Нет
Выдать	Да	Нет	Специальное	Нет
...	...	...	...	...



Визуально узлы и листья в дереве хорошо различимы: в узлах указываются правила, разбивающие содержащиеся в нем наблюдения, и производится дальнейшее ветвление. В листьях правил нет, они помечаются меткой класса, объекты которого попали в данный лист. Ветвление в листьях не производится, и они заканчивают собой ветвь дерева. Дерево решений является линейным классификатором, т.е. производит разбиение объектов в многомерном пространстве плоскостями (в двумерном случае — линиями). Если класс, присвоенный деревом, совпадает с целевым классом, то объект является распознанным, в противном случае — нераспознанным. Самый верхний узел дерева называется корневым (root node). В нем содержится весь обучающий или рабочий набор данных

Введение в деревья решений

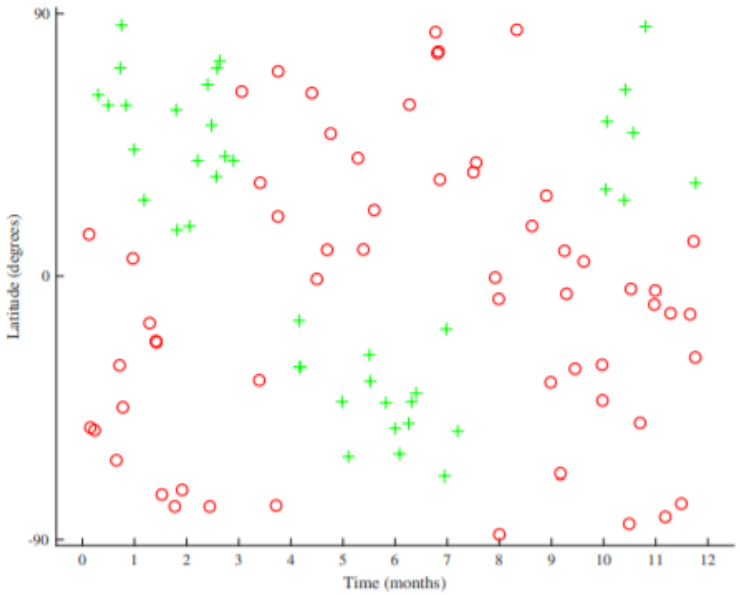


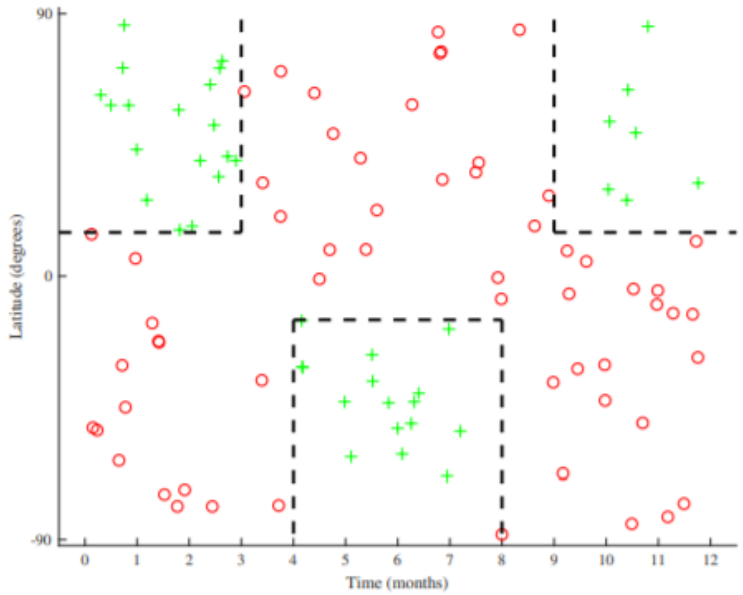
Дерево, представленное на рисунке, решает задачу классификации объектов по двум атрибутам на три класса

Введение в деревья решений

Представим, что мы хотим построить классификатор, который, учитывая время и местоположение, может предсказать, будет ли возможно кататься на лыжах поблизости.

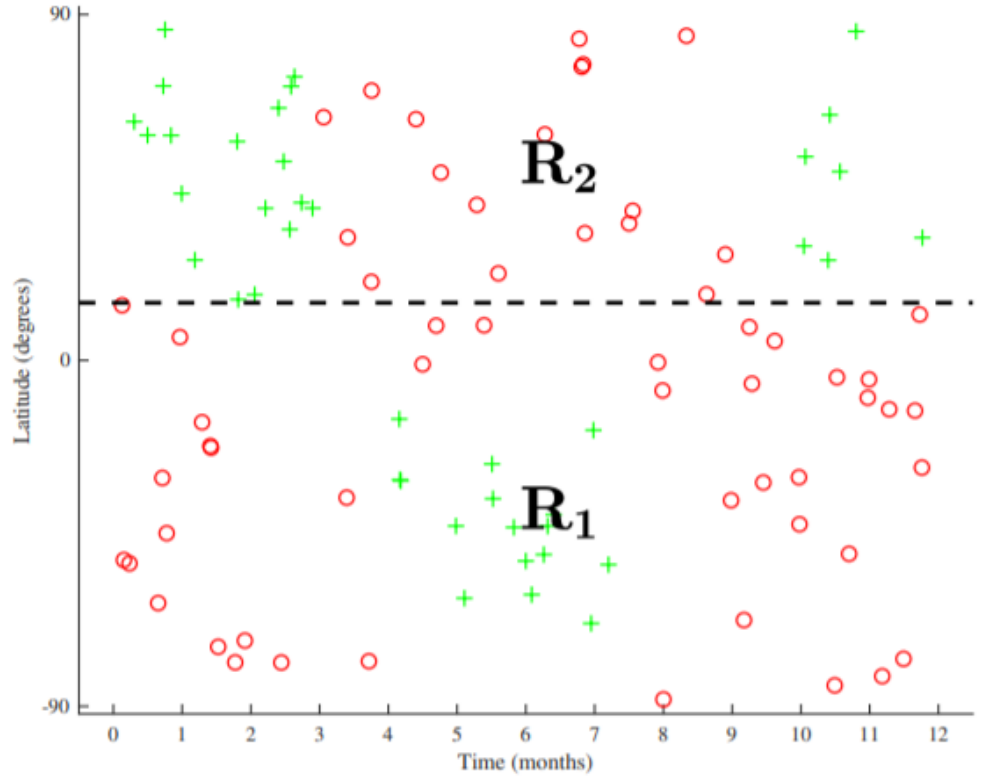
Для простоты, время представляется как месяц года, а местоположение - в виде широты (как далеко мы на севере или юге, где -90^0, 0^0 и 90^0 – это Южный полюс, Экватор и Север полюс, соответственно)



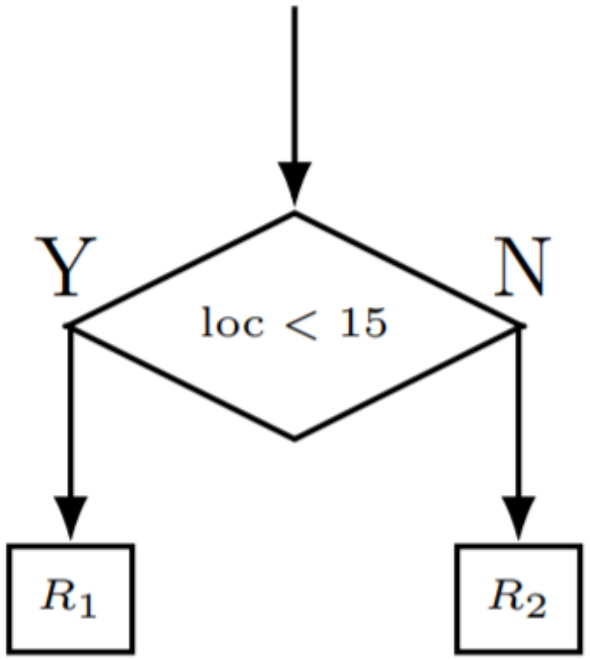


Типичный набор данных показан выше. Не существует линейной границы, которая бы правильно разделяла этот набор данных. Тем не менее, мы можем признать, что есть разные области положительного и отрицательного пространства, которые мы хотим изолировать, одно такое деление показано выше справа. Мы осуществляем это путем разбиения входного пространства X на непересекающиеся подмножества (или области) R_i

Введение в деревья решений

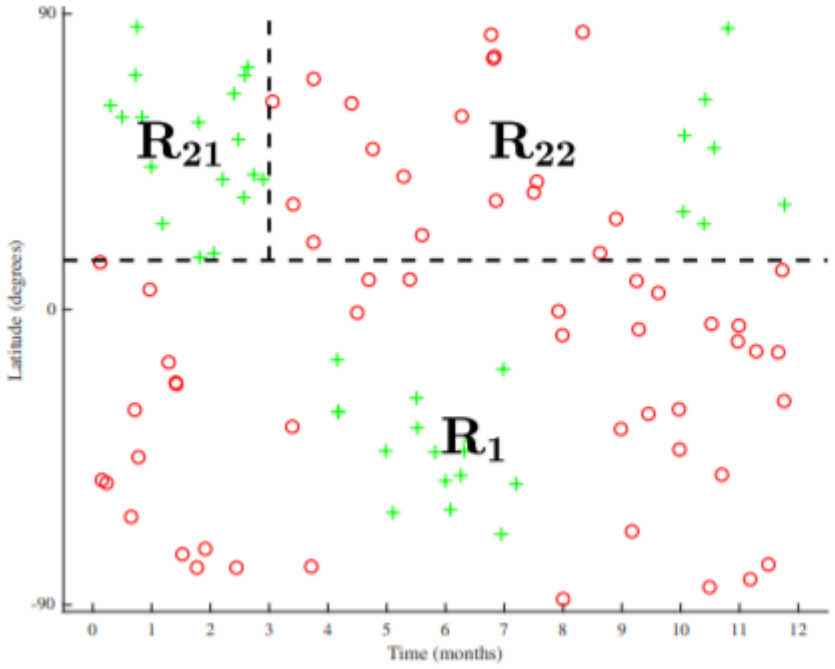


(a)

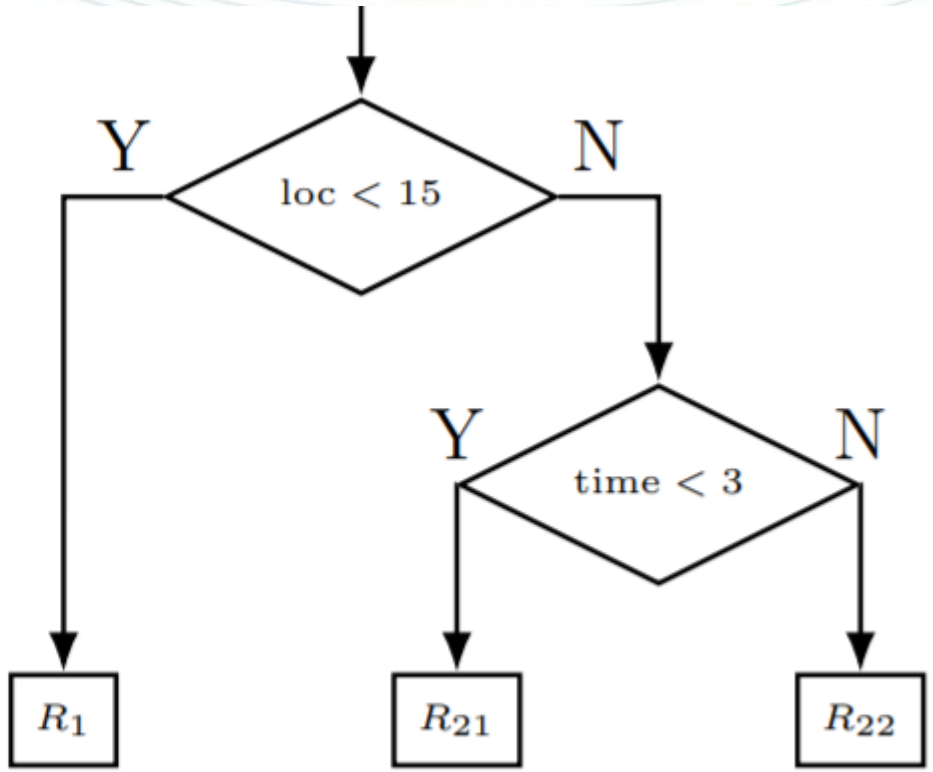


На шаге a мы разделяем входное пространство X по признаку местоположения (loc) с порогом 15, создавая дочерние области R1 и R2

Введение в деревья решений

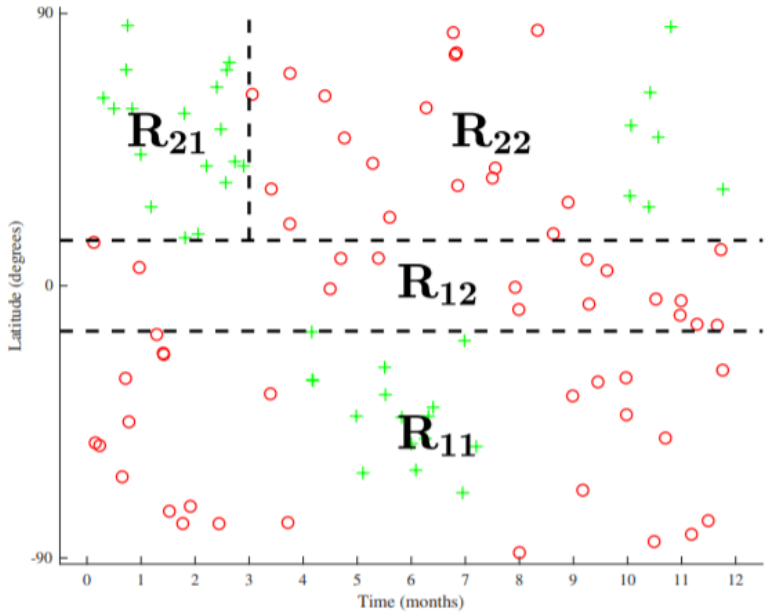


(b)

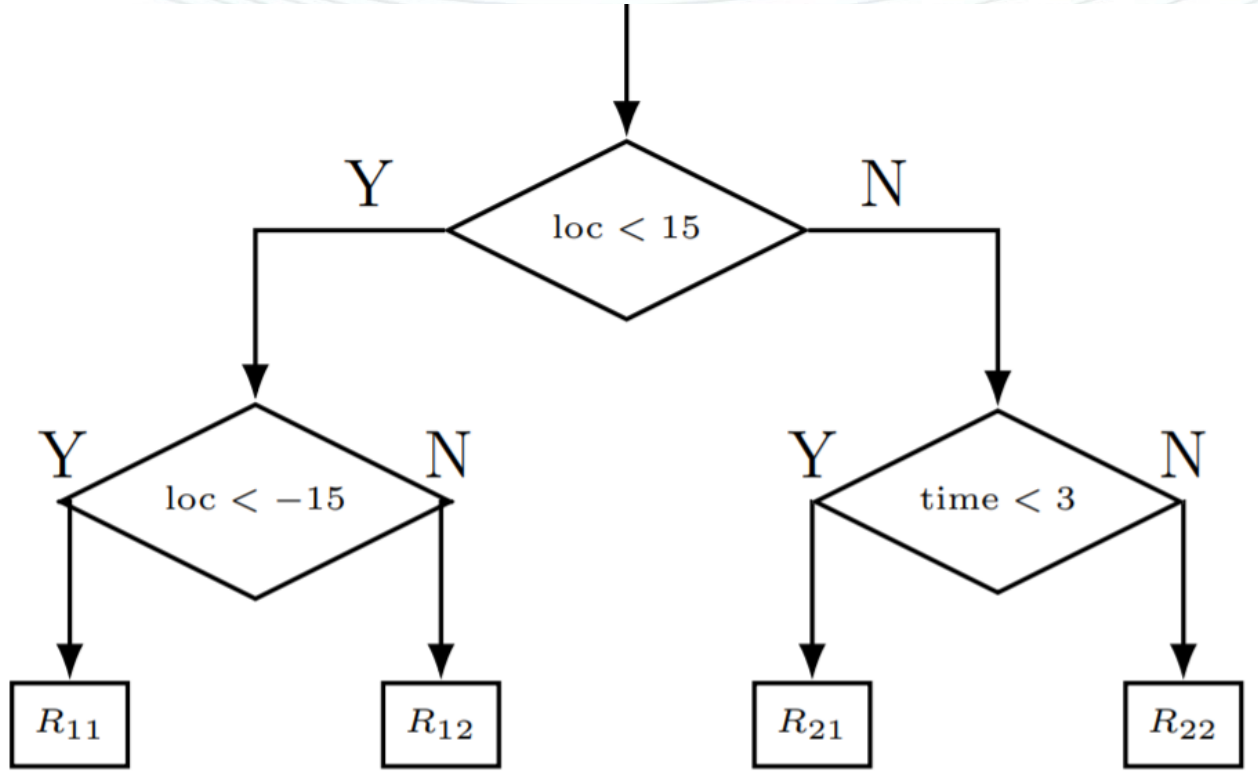


На шаге b мы затем рекурсивно выбираем одну из этих дочерних областей (в данном случае R2) и выбираем признак (время) и порог (3), создавая еще две дочерние области (R21 и R22)

Введение в деревья решений



(c)



На шаге с мы выбираем любой из оставшихся листовых узлов (R1, R21, R22). Мы можем продолжать таким образом, пока не будет выполнен заданный критерий остановки, а затем предсказать класс объектов по большинству в каждом листовом узле

Основные частные алгоритмы

- **ID3.** В основе этого алгоритма лежит понятие информационной энтропии – то есть, меры неопределенности информации (обратной мере информационной полезности величины). Для того чтобы определить следующий атрибут, необходимо подсчитать энтропию всех неиспользованных признаков относительно тестовых образцов и выбрать тот, для которого энтропия минимальна. Этот атрибут и будет считаться наиболее целесообразным признаком классификации.
- **C5.** Этот алгоритм – усовершенствование предыдущего метода, позволяющее, в частности, «усекать» ветви дерева, если оно слишком сильно «разрастается», а также работать не только с атрибутами-категориями, но и с числовыми. В общем-то, сам алгоритм выполняется по тому же принципу, что и его предшественник; отличие состоит в возможности разбиения области значений независимой числовой переменной на несколько интервалов, каждый из которых будет являться атрибутом. В соответствии с этим исходное множество делится на подмножества. В конечном итоге, если дерево получается слишком большим, возможна обратная группировка – нескольких узлов в один лист. При этом, поскольку перед построением дерева ошибка классификации уже учтена, она не увеличивается.

Основные частные алгоритмы

- **CART.** Алгоритм разработан в целях построения так называемых бинарных деревьев решений – то есть тех деревьев, каждый узел которых при разбиении «дает» только двух потомков. Грубо говоря, алгоритм действует путем разделения на каждом шаге множества примеров ровно напополам – по одной ветви идут те примеры, в которых правило выполняется (правый потомок), по другой – те, в которых правило не выполняется (левый потомок). Таким образом, в процессе «роста» на каждом узле дерева алгоритм проводит перебор всех атрибутов, и выбирает для следующего разбиения тот, который максимизирует значение показателя, вычисляемого по математической формуле и зависящего от отношений числа примеров в правом и левом потомке к общему числу примеров.