**Что такое наивный байесовский алгоритм? Какую задачу обработки данных он выполняет?**

Наивный байесовский алгоритм - это алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков.

Решает задачу классификации.НБА предполагает, что наличие признака в классе не связано с наличием какого-либо другого признака.

Например, фрукт может считаться яблоком, если он красный, круглый и его диаметр составляет порядка 8 сантиметров. Даже если эти признаки зависят друг от друга или от других признаков, в любом случае они вносят независимый вклад в вероятность того, что этот фрукт является яблоком. В связи с таким допущением алгоритм называется «наивным».

Упрощенное уравнение для классификации выглядит так:

P(Class A|Feature 1, Feature2) = P(Feature 1|Class A) \*P(Feature 2|Class A)\*P(Class A)/ P(Feature 1)\*P(Feature 2)

Уравнение находит вероятность класса А, на основании параметров 1 и 2.

Другими словами, если вы видите параметры 1 и 2, то, вероятно, это данные класса А.

Уравнение читается следующим образом: Вероятность [выявления] класса А на основании параметров 1 и 2 –это дробь.

Числитель дроби – это произведение вероятностей встретить параметр 1 и параметр 2 в классе А,

умноженная на вероятность класса А во всем наборе данных.

Знаменатель – это вероятность встретить параметр 1 и параметр 2 во всем наборе данных.

**Приведите и поясните формулу теорема Байеса.**

Теорема Байеса позволяет рассчитать апостериорную вероятность(апостериорная вероятность - это условная вероятность случайного события при условии того, что известны апостериорные данные, т. е. полученные после опыта. Она позволяет определить вероятность события после получения новой информации) P(c/x) на основе P(c), P(x) и P(x/с). Другими словами, она позволяет определить вероятность события при условии, что произошло другое статистически взаимозависимое событие. Смысл теоремы на обывательском уровне можно выразить следующим образом: теорема Байеса позволяет переставить местами причину и следствие. Зная с какой вероятностью причина приводит к некоему событию, эта теорема позволяет рассчитать вероятность того, что именно эта причина не привела к наблюдаемому событию.

Формула Байеса выглядит следующим образом:

P(c/x) = P(x/c) \* P(c) / P(x)

где:

P(c/x) - апостериорная вероятность данного класса c (т.е. данного значения целевой переменной) при данном значении признака x.

P(c) - априорная вероятность данного класса.

P(x/c) - правдоподобие, т.е. вероятность данного значения признака при данном классе.

P(x) - априорная вероятность данного значения признака.

**Деревья решений, опишите процесс работы, приемы остановки работы дерева.**

Метод деревьев решений для задач классификации состоит в том, чтобы осуществлять процесс деления исходных данных на группы, пока не будут получены однородные их множества. Совокупность правил, которые дают такое разбиение, позволят затем делать прогноз для новых данных.

Алгоритмы конструирования деревьев решений состоят из этапов:

"построение" или " создание " дерева (выбора критерия расщепления, остановки обучения (если это предусмотрено алгоритмом))

"сокращение" дерева (отсечения некоторых его ветвей)

Процесс создания дерева является нисходящим.

Алгоритм должен найти такой критерий расщепления (разбиения), чтобы разбить множество на подмножества, которые бы ассоциировались с данным узлом проверки. Каждый узел проверки должен быть помечен определенным атрибутом.

Существует правило выбора атрибута: он должен разбивать исходное множество данных таким образом, чтобы объекты подмножеств, получаемых в результате этого разбиения, являлись представителями одного класса или же были максимально приближены к нему. Это означает, что количество объектов из других классов, так называемых "примесей", в каждом классе должно стремиться к минимуму.

Качество классификационной модели, построенной при помощи дерева решений, характеризуется двумя основными признаками: точностью распознавания и ошибкой.

Остановка - такой момент в процессе построения дерева, когда следует прекратить дальнейшие ветвления.

"Ранняя остановка" - определяет целесообразность разбиения узла. Преимущество использования такого варианта - уменьшение времени на обучение модели. Но здесь возникает риск снижения точности классификации. Поэтому рекомендуется "вместо остановки использовать отсечение" (Breiman, 1984).

Ограничение глубины дерева (остановки обучения). Построение заканчивается, если достигнута заданная глубина.

Задание минимального количества примеров, которые будут содержаться в конечных узлах дерева. Ветвления продолжаются до того момента, пока все конечные узлы дерева будут содержать не более чем заданное число объектов.

На сегодняшний день существует значительное число алгоритмов, реализующих деревья решений CART, C4.5, NewId, ITrule, CHAID, CN2 и др.

**Какие типы деревьев решений вы знаете, какие индексы используются при работе дерева, для чего, что такое энтропия?**

В обучающем множестве для примеров должно быть задано целевое значение, так как деревья решений — модели, создаваемые на основе обучения с учителем. По типу переменной выделяют два типа деревьев: дерево классификации — когда целевая переменная дискретная; дерево регрессии — когда целевая переменная непрерывная.

На сегодняшний день существует значительное число алгоритмов, реализующих деревья решений CART, C4.5, NewId, ITrule, CHAID, CN2 и др.

CART (Classification and Regression Tree) – это алгоритм построения бинарного "дерева решений" – дихотомической классификационной модели. Каждый узел дерева при разбиении имеет только двух потомков. Как видно из названия алгоритма, он решает задачи классификации и регрессии; C4.5 – это алгоритм построения "дерева решений", количество потомков у узла не ограничено. Не умеет работать с непрерывным целевым полем, поэтому решает только задачи классификации.

Энтропия — это научная концепция, а также измеримое физическое свойство, которое чаще всего ассоциируется с состоянием беспорядка, случайности или неопределенности. Этот термин и понятие используются в различных областях, от классической термодинамики, где оно было впервые признано, до микроскопического описания природы в статистической физике и принципов теории информации.

Энтропия в теории информации описывает количество информации (в битах), которое необходимо, чтобы закодировать сообщение о принадлежности случайно выбранного объекта (строки) из нашей выборки X к одному из классов и передать его получателю. Если класс только один, получателю ничего не нужно передавать, энтропия равна 0. Если все классы равновероятны, то потребуется log2c бит (c – общее количество классов) – максимум функции энтропии.

E(p1,...,pn) = -Σpilog2pi

Далее, для выбора каждого аттрибута А вычисляется индекс GAIN

Gain(X,A) = E(X)-Σ |Xa| / |X|\*E(Xa), a принадлежит values(A)

Так же, есть индекс GINI, и алгоритм Random Forest - он создает множество деревьев приятия решений и потом усредняет результат их предсказаний.

Индекс gini это вот если 6\_2 делали, там на дереве отметки есть. По сути, просто a>5, a<5. Индекс расщепления

Когда в дереве происходит расщепление, то всегда берется наибольший по энтропии параметр, потом второй и так далее.

**Назовите достоинства и недостатки деревьев решения.**

Основными достоинствами метода "деревья решений" являются:

быстрый процесс обучения;генерация правил в областях, где эксперту трудно формализовать свои знания;извлечение правил на естественном языке;понятная на интуитивном уровне классификационная модель;высокая точность прогноза, сопоставимая с другими методами (статистика, нейронные сети);построение непараметрических моделей.

В силу этих и многих других причин, методология "деревьев решений" является важным инструментом в работе каждого специалиста, занимающегося анализом данных, вне зависимости от того, практик он или теоретик.

Недостатки использования деревьев решений:

Сложно получить оптимальное дерево;Проблема с переобучением;Иногда сложно объяснить решение (если дерево большое и запутанное)Сложности с количественными задачами (если прецедент находится на стыке классов – к какому из классов его правильнее отнести?)

**Принцип работы RandomForest.**

Random Forest - один из наиболее популярных алгоритмов машинного обучения или data mining.

Во-первых он невероятно универсален, с его помощью можно решать как задачи регрессии так и классификации. Проводить поиск аномалий и отбор предикторов.

Во-вторых это тот алгоритм, который действительно сложно применить неправильно. Просто потому, что в отличии от других алгоритмов у него мало настраиваемых параметров. И еще он удивительно прост по своей сути. И в то же время он отличается высокой точностью.

Главная идея алгоритма - обучение ансамбля в действии. Алгоритм Random Forest потому и называется "Случайный Лес", что для полученных данных он создает множество деревьев принятия решений и потом усредняет результат их предсказаний. Важным моментом тут является элемент случайности в создании каждого дерева - если мы создадим много одинаковых деревьев с разной точностью, то результат их усреднения будет обладать точностью одного дерева.

Пример реализации

> library("randomForest")

> rf <- randomForest(Species ~ .,

+ data=trainData, ntree=100, proximity=TRUE)

> table(predict(rf), trainData$Species)