1. Что такое случайный лес?

Random Forest — это мощный и универсальный алгоритм машинного обучения с учителем, который наращивает и объединяет несколько деревьев решений для создания «леса». Его можно использовать как для задач классификации, так и для регрессии в R и Python.

Что такое контролируемое машинное обучение?

Контролируемое машинное обучение — это когда алгоритм (или модель) создается с использованием так называемого обучающего набора данных. Модель обучается с использованием множества различных примеров различных входных и выходных данных и, таким образом, учится классифицировать любые новые входные данные, которые она получит в будущем. Вот как алгоритмы используются для прогнозирования будущих результатов.

Что такое регрессия и классификация в машинном обучении?

В машинном обучении алгоритмы используются для классификации определенных наблюдений, событий или входных данных в группы. Например, спам-фильтр электронной почты будет классифицировать каждое электронное письмо как «спам» или «не спам». Однако пример электронной почты очень прост; в бизнес-контексте предсказательные способности таких моделей могут оказать большое влияние на то, как принимаются решения и как формируются стратегии, но об этом позже.

Итак: регрессия и классификация — это задачи контролируемого машинного обучения, используемые для прогнозирования значения или категории результата или результата.

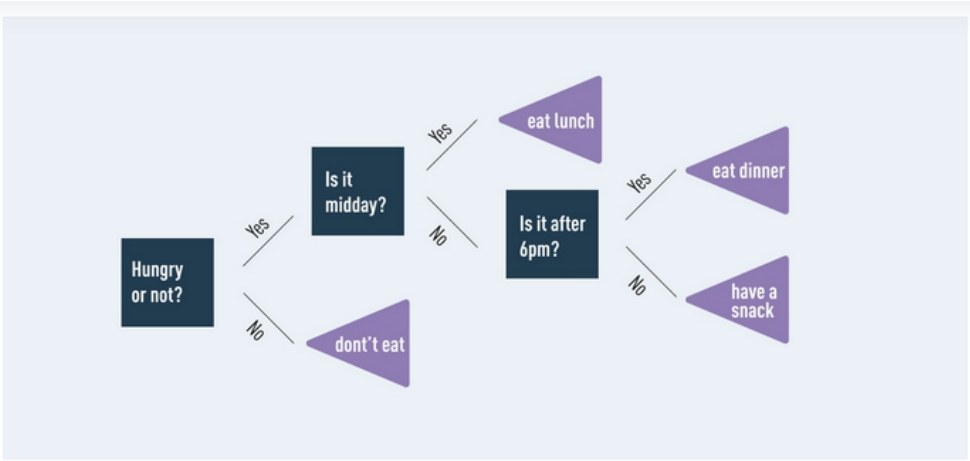
В классификационном анализе зависимый атрибут является категориальным. Задачи классификации учат, как присваивать метку класса примерам из предметной области. Как упоминалось ранее, распространенным примером классификации является спам-фильтр вашей электронной почты.

В регрессионном анализе зависимый атрибут является числовым. Регрессия используется, когда выходная переменная представляет собой реальное или непрерывное значение, такое как зарплата, возраст или вес.

Для простого различения между ними помните, что классификация предназначена для прогнозирования метки (например, «спам» или «не спам»), а регрессия — для прогнозирования количества.

Что такое деревья решений?

Как мы знаем, модель случайного леса растет и объединяет несколько деревьев решений для создания «леса». Дерево решений — это еще один тип алгоритма, используемого для классификации данных. Проще говоря, вы можете думать об этом как о блок-схеме, которая показывает четкий путь к решению или результату; он начинается в одной точке, а затем разветвляется в двух или более направлениях, причем каждая ветвь дерева решений предлагает разные возможные результаты.



Классификация — важная и очень ценная отрасль науки о данных, а случайный лес — это алгоритм, который можно использовать для таких задач классификации. Ансамбль деревьев Random Forest выводит моду или среднее значение отдельных деревьев. Этот метод позволяет получать более точные и стабильные результаты, полагаясь на множество деревьев, а не на одно дерево решений. Это как разница между одноколесным велосипедом и квадроциклом!

2. Как работает алгоритм случайного леса?

Random Forest создает несколько деревьев решений, которые объединяются для более точного прогноза.

Логика модели случайного леса заключается в том, что несколько некоррелированных моделей (отдельных деревьев решений) работают гораздо лучше в группе, чем по отдельности. При использовании Random Forest для классификации каждое дерево дает классификацию или «голосование». Лес выбирает классификацию с большинством «голосов». При использовании Random Forest для регрессии лес выбирает среднее значение результатов всех деревьев.

Ключевым моментом здесь является тот факт, что корреляция между отдельными моделями, то есть между деревьями решений,

составляющими более крупную модель случайного леса, низка (или отсутствует). В то время как отдельные деревья решений могут давать ошибки, большая часть группы будет правильной, что приведет к общему результату в правильном направлении.

Отличаются ли деревья решений в Random Forest от обычных деревьев решений?

Легко запутаться в одном дереве решений и лесе решений. Кажется, что лес решений — это набор отдельных деревьев решений, и это… что-то вроде. Это набор отдельных деревьев решений, но все деревья смешиваются случайным образом, а не отдельные деревья, растущие по отдельности.

При использовании обычного дерева решений вы должны ввести обучающий набор данных с функциями и метками, и он сформулирует некоторый набор правил, которые он будет использовать для прогнозирования. Если вы введете ту же информацию в алгоритм случайного леса, он случайным образом выберет наблюдения и признаки для построения нескольких деревьев решений, а затем усреднит результаты.

Например, если вы хотите предсказать, как часто клиент банка будет использовать конкретную услугу, предоставляемую банком, с помощью единого дерева решений, вы должны собрать, как часто они пользовались банком в прошлом и какие услуги они использовали во время своих визитов. . Вы бы добавили некоторые функции, которые описывают решения этого клиента. Дерево решений будет генерировать правила, которые помогут предсказать, будет ли клиент пользоваться услугами банка.

Если вы введете тот же набор данных в случайный лес, алгоритм построит несколько деревьев из случайно выбранных посещений клиентов и использования услуг. Затем он выводит средние результаты каждого из этих деревьев.

Как тренируются деревья в случайном лесу?

Деревья решений в ансамбле, как и деревья в случайном лесу, обычно обучаются с использованием метода «мешков». Метод «бэгинга» — это разновидность ансамблевого алгоритма машинного обучения, называемого агрегацией Bootstrap. Метод ансамбля объединяет прогнозы нескольких алгоритмов машинного обучения вместе, чтобы делать более точные прогнозы, чем индивидуальная модель. Случайный лес также является ансамблевым методом.

Bootstrap случайным образом выполняет выборку строк и признаков из набора данных, чтобы сформировать образцы наборов данных для каждой модели. Агрегирование сводит эти выборочные наборы данных в сводную статистику на основе наблюдения и объединяет их.

Bootstrap Aggregation можно использовать для уменьшения дисперсии алгоритмов с высокой дисперсией, таких как деревья решений.

Дисперсия — это ошибка, возникающая из-за чувствительности к небольшим колебаниям в наборе данных, используемом для обучения. Высокая дисперсия заставит алгоритм моделировать нерелевантные данные или шум в наборе данных вместо предполагаемых выходных данных, называемых сигналом. Эта проблема называется переоснащением. Переобученная модель будет хорошо работать при обучении, но не сможет отличить шум от сигнала в реальном тесте.

Бэггинг — это применение метода начальной загрузки к алгоритму машинного обучения с высокой дисперсией.

4. Каковы преимущества Random Forest?

Random Forest популярен, и не зря! Он предлагает множество преимуществ, от точности и эффективности до относительной простоты использования. Для специалистов по данным, желающих использовать случайные леса в Python, scikit-learn предлагает простую и эффективную библиотеку классификатора случайных лесов.

Наиболее удобным преимуществом использования случайного леса является его способность по умолчанию корректировать привычку деревьев решений к переоснащению их обучающей выборки. Использование метода мешков и случайного выбора признаков при выполнении этого алгоритма почти полностью решает проблему переобучения, что очень хорошо, поскольку переобучение приводит к неточным результатам. Кроме того, даже если некоторые данные отсутствуют, Random Forest обычно сохраняет свою точность.

Случайный лес намного эффективнее одного дерева решений при выполнении анализа большой базы данных. С другой стороны, случайный лес менее эффективен, чем нейронная сеть. Нейронная сеть, иногда называемая просто нейронной сетью, представляет собой серию алгоритмов, которые выявляют лежащую в основе взаимосвязь в наборе данных, имитируя способ мышления человеческого мозга.

Нейронные сети более сложны, чем случайные леса, но дают наилучшие возможные результаты, адаптируясь к изменяющимся входным данным. В отличие от нейронных сетей, Random Forest настроен таким образом, чтобы обеспечить быструю разработку с минимальными гиперпараметрами (высокоуровневыми архитектурными рекомендациями), что сокращает время настройки.

Поскольку для разработки случайного леса требуется меньше времени и опыта, этот метод часто перевешивает долгосрочную эффективность нейронной сети для менее опытных специалистов по данным.

Итак, резюмируя, ключевыми преимуществами использования Random Forest являются:

1. Простота использования
2. Эффективность
3. Точность
4. Универсальность – может использоваться для классификации или регрессии
5. Более удобен для начинающих, чем такие же точные алгоритмы, как нейронные сети.

5. Каковы недостатки Random Forest?

У Random Forest не так много недостатков, но у каждого инструмента есть свои недостатки. Поскольку случайный лес использует много деревьев решений, для больших проектов может потребоваться много памяти. Это может сделать его медленнее, чем некоторые другие, более эффективные алгоритмы.

Иногда, поскольку это метод, основанный на дереве решений, а деревья решений часто страдают от переобучения, эта проблема может повлиять на весь лес. Эта проблема обычно предотвращается случайным лесом по умолчанию, поскольку он использует случайные подмножества функций и строит меньшие деревья с этими подмножествами. Это может замедлить скорость обработки, но повысить точность.