Шпаргалка по Random forest / Концепции Cheatsheet (XeLaTeX)

Краткий справочник

Содержание

Случайный Лес: Определение

Случайный Лес (Random Forest, RF) — это ансамблевый метод машинного обучения, который строит множество деревьев решений во время обучения и выводит класс, который является модой классов (классификация) или средним предсказанием (регрессия) отдельных деревьев. Это один из самых популярных и эффективных "из коробки" алгоритмов.

Аналогия: Комитет Экспертов

Представьте, что для принятия важного решения вы собираете **комитет разных экспертов** (много деревьев). Каждый эксперт смотрит на проблему немного под своим углом. Вы принимаете решение на основе их коллективного мнения. Случайный лес применяет тот же принцип, но с деревьями решений.

1 Идея Бэггинга (Bagging)

Определение: Bagging

Бэггинг — это основной принцип, лежащий в основе Случайного Леса. Он сочетает две ключевые идеи: Bootstrap и Aggregating.

Шаг 1: Bootstrap (Бутстрэп)

Создается множество (N) подвыборок из исходного обучающего датасета. Каждая подвыборка формируется путем случайного выбора объектов **с возвращением**.

- Некоторые объекты могут попасть в одну подвыборку несколько раз.
- Некоторые объекты могут не попасть ни разу.
- Размер каждой подвыборки обычно равен размеру исходного датасета.

Объекты, не попавшие в конкретную бутстрэп-выборку ($\approx 37\%$), называются **Out-of-Bag (OOB)**.

Шаг 2: Aggregating (Агрегация)

На каждой Bootstrap-подвыборке независимо обучается своя модель (в случае RF — дерево решений). Затем предсказания всех N моделей агрегируются:

- **Регрессия:** Усреднение предсказаний.
- **Классификация:** Голосование большинством (выбор самого популярного класса).

Цель Бэггинга и OOB-оценка

Цель бэггинга: Снизить **дисперсию (variance)** ансамблевой модели. Индивидуальные деревья могут иметь высокую дисперсию (переобучаться), но усреднение их предсказаний сглаживает ошибки и повышает устойчивость.

OOB-оценка: Объекты Out-of-Bag могут использоваться для оценки качества модели (OOB-оценка) без необходимости выделения отдельной валидационной выборки. Для каждого OOB-объекта предсказание делается ансамблем деревьев, которые *не обучались* на этом объекте.

2 Случайный Выбор Признаков (Feature Subsampling)

Дополнительная Случайность в RF

В отличие от простого бэггинга деревьев, Случайный Лес вносит **дополнительный элемент случайности** при построении каждого дерева:

- При поиске лучшего разбиения (split) в каждом узле дерева, алгоритм рассматривает не все доступные признаки, а только их **случайное подмножество**.
- Размер этого подмножества (`max_features`) является важным гиперпараметром.
- Типичные значения `max_features`: $\sqrt{p}$ для классификации, $p/3$ для регрессии (где p — общее число признаков).

Цель Случайного Выбора Признаков: Декорреляция Деревьев

Зачем это нужно? Это делается для **декорреляции** деревьев в ансамбле.

- Если бы все деревья видели все признаки, и существовал бы один очень сильный признак, большинство деревьев использовали бы его для первого (и, возможно, последующих) разбиений.
- В результате деревья были бы очень похожими (скоррелированными).
- Усреднение предсказаний сильно скоррелированных моделей не дает значительного снижения дисперсии.

Случайный выбор признаков заставляет разные деревья фокусироваться на разных наборах признаков, делая их более **разнообразными** и независимыми.

Аналогия: Разные Аспекты для Экспертов

Возвращаясь к комитету экспертов: чтобы они не пришли к одному выводу, опираясь на самый очевидный факт, вы просите каждого эксперта при анализе сосредоточиться только на **случайном наборе аспектов** проблемы. Это побуждает их исследовать разные стороны вопроса и дает более надежный коллективный результат.

3 Как Уменьшает Дисперсию

Борьба с Переобучением через Усреднение

Ключевая сила Случайного Леса — в его способности значительно **уменьшать дисперсию** по сравнению с одним деревом решений, при этом не сильно увеличивая (или даже немного уменьшая) **смещение (bias)**.

Сравнение Bias/Variance: Одно Дерево vs RF

- **Одно дерево решений:**
  - *Низкое смещение:* Может хорошо подогнаться под обучающие данные, уловить сложные зависимости.
  - *Высокая дисперсия:* Сильно меняется при небольшом изменении данных, легко переобучается.

- **Случайный Лес (RF):**
  - *Относительно низкое смещение:* Наследует гибкость от деревьев.
  - *Значительно сниженная дисперсия:* Благодаря усреднению и декорреляции.

Механизмы Снижения Дисперсии в RF

- **Бэггинг (усреднение):** Усреднение предсказаний N моделей, ошибки которых не полностью скоррелированы, приводит к снижению общей дисперсии ансамбля. Чем больше деревьев (N), тем ниже дисперсия (до определенного предела).
- **Случайный выбор признаков (декорреляция):** Уменьшает корреляцию между деревьями, что делает усреднение еще более эффективным для снижения дисперсии.

В итоге, RF достигает хорошего **компромисса смещения-дисперсии (Bias-Variance Tradeoff)**, создавая модель, которая одновременно гибкая и устойчивая к переобучению.

Аналогия: Мудрость Толпы

Один человек может сильно ошибаться в оценке (высокая дисперсия). Но если усреднить оценки большой группы людей, где ошибки случайны и не связаны, итоговая оценка будет гораздо ближе к истине (низкая дисперсия). RF использует похожий принцип.

4 Важность Признаков (Feature Importance)

Зачем Оценивать Важность?

Хотя Случайный Лес — это ансамбль, что усложняет прямую интерпретацию, он предоставляет методы для оценки вклада каждого признака. Это помогает:

- Понять, какие данные действительно влияют на модель.
- Упростить модель через отбор признаков (Feature Selection).
- Получить инсайты о предметной области.

Существует два основных подхода: MDI (Gini Importance) и Permutation Importance (MDA).

4.1. A Mean Decrease in Impurity (MDI) / Gini Importance

Идея MDI: Вклад в Чистоту Узлов

Этот метод оценивает важность признака на основе того, насколько сильно его использование для разделений в деревьях **уменьшает нечистоту (Impurity)** узлов (например, Gini Impurity для классификации или MSE для регрессии). Признак считается важным, если он часто выбирается для разделения и эти разделения значительно "очищают" данные. Расчет происходит **на обучающей выборке** во время построения леса.

Как Считается MDI (Детально)

1. Обучаем Random Forest.
2. Для **каждого дерева** в лесу:
- Для **каждого внутреннего узла**, где произошло разделение по признаку F:
  - Рассчитываем **уменьшение нечистоты (Information Gain / Variance Reduction)** в этом узле: $\Delta Impurity_{node} = Impurity(parent) - WeightedImpurity(children)$.
  - Рассчитываем **взвешенное уменьшение нечистоты**: $WeightedDecrease_{node} = N_{node} \times \Delta Impurity_{node}$, где N_{node} — количество объектов в узле.
3. Для **каждого признака** F:
- Суммируем взвешенные уменьшения нечистоты ($WeightedDecrease_{node}$) по **всем узлам всех деревьев**, где признак F использовался для разделения. Это дает "общую важность" $TotalImportance(F)$.
4. **Нормализация:** Общую важность каждого признака делят на сумму важностей всех признаков: $Importance(F) = \frac{TotalImportance(F)}{\sum_j TotalImportance(F_j)}$.

Формула (концептуально):

$$Importance_{MDI}(F) \propto \sum_{\text{trees}} \sum_{\substack{\text{nodes split} \\ \text{on } F}} N_{\text{node}} \cdot \Delta Impurity_{\text{node}}$$

Плюсы MDI

- **Быстро** считается (информация доступна сразу после обучения).
- Обычно предоставляется по умолчанию в библиотеках (например, `feature_importances_` в scikit-learn).

Минусы и Предостережения по MDI

- Склонен **завышать важность** числовых признаков и категориальных признаков с большим количеством уникальных значений (высокой кардинальностью).
- Может давать **неадекватные результаты для скоррелированных признаков** (важность может "делиться" между ними или присваиваться только одному).

- Показывает, насколько признак был *полезен для построения деревьев* на обучающих данных, но не обязательно, насколько он важен для *предсказаний* на новых данных. **Использовать с осторожностью!**

4.2. B Mean Decrease in Accuracy (MDA) / Permutation Importance

Идея MDA: Влияние "Поломки" Признака на Качество

Этот метод оценивает важность признака, измеряя, насколько **ухудшится качество предсказания** модели (например, Accuracy, F1, R^2, MSE), если "сломать" связь между этим признаком и целевой переменной путем случайного перемешивания его значений. Расчет происходит **на отложенной (не обучающей!) выборке**.

Как Считается Permutation Importance (Детально)

1. Обучаем Random Forest.
2. Выбираем **отложенную выборку** (OOB, валидационную или тестовую).
3. Рассчитываем **базовую метрику качества** $Score_{base}$ модели на этой выборке.
4. Для **каждого признака** F:
- Создаем копию отложенной выборки.
- В этой копии **случайно перемешиваем значения** только в столбце признака F.
- Делаем предсказания модели на **модифицированной** выборке.
- Рассчитываем метрику качества $Score_{permuted}(F)$ на этих предсказаниях.
- **Важность признака** F = $Score_{base} - Score_{permuted}(F)$.
5. (Опционально, для стабильности) Повторяем шаг 4 несколько раз с разными случайными перемешиваниями для каждого признака и усредняем полученные значения важности.

Формула (концептуально):

$$Importance_{Permutation}(F) = Score_{base} - \mathbb{E}[Score_{permuted}(F)]$$

где $\mathbb{E}[\cdot]$ означает ожидаемое значение по разным перемешиваниям.

Плюсы Permutation Importance

- Более **надежен**, чем MDI, как показатель реального влияния на предсказания.
- Напрямую измеряет влияние признака на **предсказательную способность** модели на новых данных.
- Идея метода **модель-агностична** (можно применять к любой обученной модели).

Минусы и Предостережения по Permutation Importance

- **Вычислительно затратен** (требует многократных предсказаний модели).
- Результат может зависеть от конкретной отложенной выборки и случайности перемешивания (рекомендуется усреднять по нескольким запускам).
- Интерпретация при **сильно скоррелированных признаках** требует осторожности: перемешивание одного признака может не сильно ухудшить метрику, если модель может использовать его коррелированный "заменитель". Это может привести к занижению важности обоих признаков.

4.3. C Сравнение MDI и Permutation Importance

Ключевые Различия (Частый Вопрос на Собеседованиях)

Характеристика	MDI (Gini Importance)
Что измеряет?	Насколько признак **использовался** для уменьшения нечистоты узлов при **обучении**.
На каких данных?	Обучающая выборка
Скорость	Быстро
Надежность	Менее надежен, предвзят к типу признаков, обучающей выборке
Скоррел. признаки	Может "делить", завышать/занижать важность
Модель-агностичность	Специфичен для деревьев
Основное Применение	Быстрый анализ, оценка по умолчанию

Ключевой вывод: Permutation Importance обычно считается более надежным показателем реальной важности признака для **производительности** модели. MDI показывает "популярность" признака при построении модели.

5 Ключевые Гиперпараметры

Основные Параметры для Настройки RF

Хотя RF часто хорошо работает "из коробки", тюнинг гиперпараметров может улучшить результат. Важнейшие из них:

`n_estimators`

Количество деревьев в лесу.

- **Влияние:** Больше деревьев -> ниже дисперсия ансамбля (до некоторого предела), стабильнее результат, но дольше обучение и предсказание.
- **Типичные значения:** 100, 500, 1000 и более. Обычно выбирают достаточно большим значением, пока производительность на валидации не перестанет расти или время обучения не станет чрезмерным.

`max_features`

Количество признаков, случайно выбираемых для рассмотрения при поиске лучшего сплита в каждом узле.

- **Влияние:** Ключевой параметр для контроля корреляции между деревьями. Меньшее значение -> более декоррелированные деревья -> большее снижение дисперсии, но потенциально большее смещение (каждое дерево слабее). Большее значение -> деревья более похожи -> меньшее снижение дисперсии, но потенциально меньшее смещение.
- **Типичные значения (и отправные точки):** `sqrt(p)` (классификация), `p/3` или `log2(p)` (регрессия). Часто требует подбора через кросс-валидацию.

Параметры Отдельных Деревьев

Гиперпараметры базовых деревьев решений также влияют на лес и могут использоваться для контроля сложности и предотвращения переобучения, хотя RF менее чувствителен к ним, чем одно дерево.

- **`max_depth`:** Максимальная глубина деревьев. Ограничение глубины уменьшает сложность и дисперсию отдельных деревьев.
- **`min_samples_split`:** Минимальное количество объектов в узле для его разделения.
- **`min_samples_leaf`:** Минимальное количество объектов в листовом узле.

Стратегия: Часто в RF деревья строят почти до максимальной глубины (e.g., `max_depth=None`), полагаясь на усреднение и `max_features` для контроля переобучения. Однако ограничение глубины или увеличение `min_samples_leaf`/`min_samples_split` может быть полезно для уменьшения размера модели и времени обучения, иногда даже улучшая качество.

6 Сравнение с Конкурентами

6.1. A RF vs. Одно Дерево Решений

Преимущества RF перед Одним Деревом

- Значительно **меньше переобучается** благодаря усреднению и декорреляции.
- Гораздо **более устойчив** к изменениям в данных (низкая дисперсия).
- Обычно показывает **более высокую точность** и обобщающую способность на практике.

Недостатки RF перед Одним Деревом

- **Менее интерпретируем.** Сложно понять логику принятия решения ансамбля по сравнению с одним деревом, которое можно визуализировать.
- Требует **больше вычислительных ресурсов** (память для хранения деревьев, время для обучения и предсказания).

6.2. B RF vs. Линейные Модели (Логистическая/-Линейная Регрессия)

Преимущества RF перед Линейными Моделями

- Легко улавливает **нелинейные зависимости** между признаками и целью.
- Автоматически обрабатывает **взаимодействия** между признаками.
- **Не требует масштабирования** признаков (решения в узлах основаны на порогах).
- Менее чувствителен к **выбросам** в признаках.
- Часто дает хорошее качество **"из коробки"** с минимальной предобработкой данных и настройкой.

Недостатки RF перед Линейными Моделями

- **Менее интерпретируем**, чем линейные модели, где веса имеют ясный смысл (при условии корректной подготовки данных).
- Может быть **медленнее** в обучении и особенно в предсказании на очень больших датасетах или при большом количестве деревьев.
- Плохо **экстраполирует**. Предсказания RF ограничены диапазоном значений целевой переменной, виденных в обучающих данных (по сути, среднее по листьям). Линейные модели могут экстраполировать.
- Может требовать **значительно больше памяти**.
- На **очень разреженных данных** (много нулей, как в тексте)

  линейные модели часто работают лучше и быстрее.