Математические Методы Прогнозирования

Кафедра Интеллектуальных Систем

Отчёты о лабораторной работе №1

Оглавление

Page

Глава 1

1.1. Введение

Модель Lightgbm, которая была использована в данной работе, основана на базовом алгоритме Градиентного Бустинга над Решающими Деревьями (GBDT). Алгоритм GBDT использует деревья решений (decision trees) в качестве основных эстиматоров, причем деревья применяются не как отдельные независимые алгоритмы, а как последовательности алгоритмов, которые используются в сочетании — результат ошибок первой модели (residuals) подаются в последующую модель и используются для ее тренировки. Процедура повторяется далее и итоговый прогноз модели базируется на сумме предсказаний всех моделей. Описание данного процесса можно найти на рис.1.1

Пусть $\{x_i, y_i\}_{i=1}^m$— признаковое множество модели, где $x_i = (x_{1i}, x_{2i}, \cdots, x_{ri})$ — признаки модели, а y_i — целевая переменная. Тогда мы можем описать работу алгоритма GBDT, как последовательность шагов:

- Шаг 1: Изначальное константное значение γ задается как

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{m} L\left(y_i, \gamma\right),$$

  где L — функция потерь
- Шаг 2: Ошибка по направлению градиента записывается как

$$\hat{y}_i = -\left[\frac{\partial L\left(y_i, F\left(x_i\right)\right)}{\partial F\left(x_i\right)}\right]_{f(x)=f_{n-1}(x)},$$

  где n - количество итераций
- Шаг 3: Первоначальная модель тренируется на входных данных и параметр α_n рассчитывается методом наименьших квадратов:

$$\alpha_n = \arg\min_{\alpha,\beta} \sum_{i=1}^{m} \left(\hat{y}_i - \beta T\left(x_i; \alpha\right)\right)^2$$

- Шаг 4: Минимизируя функцию потерь получаем веса γ_n:

$$\gamma_n = \arg\min_{\gamma} \sum_{i=1}^{m} L\left(y_i, F_{n-1}(x) + \gamma T\left(x_i; \alpha_n\right)\right)$$

- Шаг 5: Модель обновляется следующим образом до момента сходимости либо до завершения итераций:

$$F_n(x) = F_{n-1}(x) + \gamma_n T\left(x_i; \alpha_n\right)$$

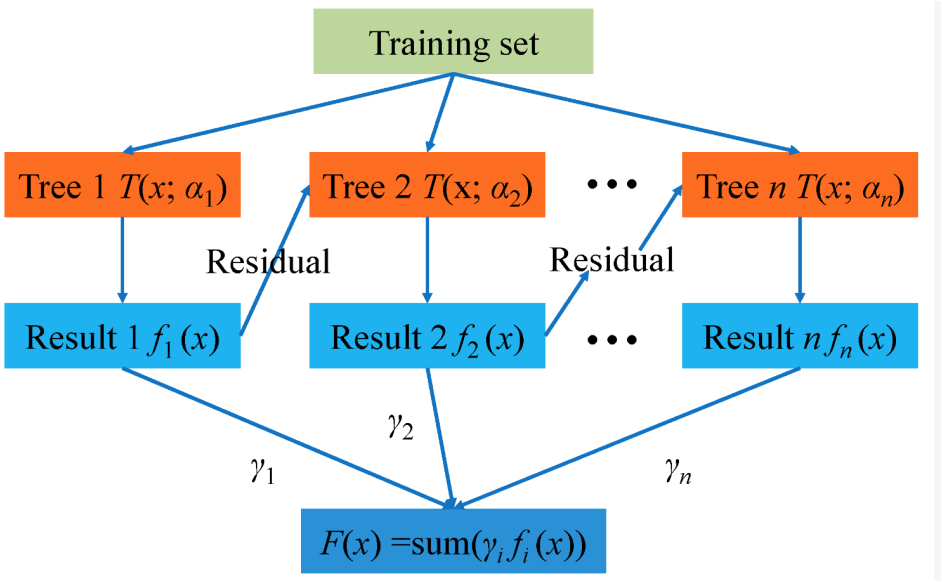

Рис. 1.1. "Модель работы алгоритма GBDT"

LightGBM — алгоритм, разработанный Microsoft Research на основе GBDT. Его цель - повысить эффективность вычислений, чтобы более успешно решать проблемы прогнозирования на основе больших данных. В алгоритме GBDT для отбора и разбиения показателей используется метод предварительной сортировки. Хотя этот метод может точно определить точку разбиения, он требует больше времени и памяти. В LightGBM для увеличения скорости обучения и снижения потребления памяти используется алгоритм на основе гистограмм и стратегия роста деревьев по листьям с ограничением максимальной глубины.

Стратегии роста по уровням и по листьям показаны на рис.1.2.

Согласно стратегии роста по уровням, листья на одном слое одновременно разделяются. Это позволяет оптимизировать работу с несколькими потоками и контролировать сложность модели. Однако листья на одном и том же слое обрабатываются без разбора, в то время как они имеют разный информационный выигрыш (Information Gain). Информационный выигрыш указывает на ожидаемое снижение энтропии, вызванное разделением узлов на основе атрибутов, которое можно определить следующим образом:

$$IG(B, V) = En(B) - \sum_{v \in \text{Values}(V)} \frac{|B_v|}{B} \text{En}\,(B_v)$$

$$\text{En}(B) = \sum_{d=1}^{D} -p_d \log_2 p_d,$$

$En(B)$ — это информационная энтропия коллекции B, D — количество категорий, v — значение атрибута V и Bv — это подмножество B для которого значение

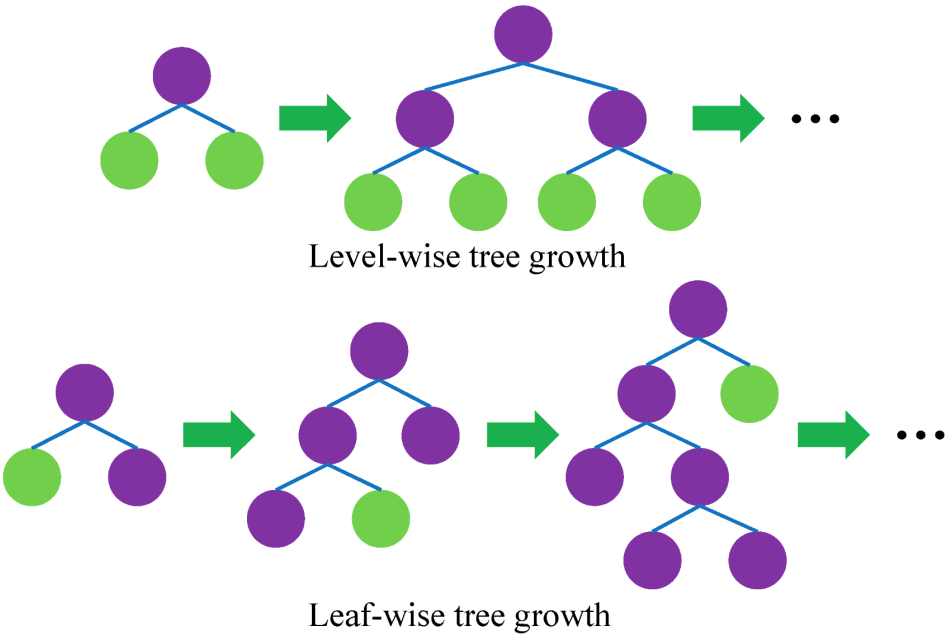

Рис. 1.2. "Модель работы алгоритма LightGBM"

атрибута равно v.

1.2. Постановка задачи

В данной работе, мы будем оперировать с авторегрессионной моделью. Это модель временного ряда, в которой его текущее значение линейно зависит от предыдущих (ретроспективных) значений этого же ряда. Линейная зависимость означает, что текущее значение равно взвешенной сумме нескольких предыдущих значения ряда:

$$Y(t) = C + b_1 Y(t-1) + \ldots + b_n(t-n) + \varepsilon(t) = C + \sum_{i=1}^{n} b_i Y_{t_0-n} + \varepsilon_t,$$

где C - константа, которую для простоты часто полагают равной 0; n - число ретроспективных значений ряда, b - коэффициенты (параметры) модели, ε_t - случайная составляющая.

Если временной ряд представляет собой ежедневные показатели, то $Y(t)$ - показатели на сегодня, $Y(t-1)$ - показатели, которые были вчера, $Y(t-2)$ - позавчера и т.д. Возможно использование и других шкал наблюдений — еженедельные, ежеквартальные и т.д.

С помощью авторегрессионного подхода также можно производить дальнейший анализ временных рядов — выявлять корреляции, сезонность и другие особенности.

1.3. Модель

Чтобы сделать прогноз с помощью LightGBM, нам нужно сначала преобразовать данные временного ряда в табличный формат, где признаки создаются с лагом (т.е. $y-1, y-2, y-3$, ...). Поскольку модель может предсказать только одношаговый прогноз, предсказанное значение используется для характеристики на следующем шаге, когда мы создаем многошаговое прогнозирование (multi—step forecasting), что называется рекурсивным подходом для многошагового прогнозирования (см. рис.1.3).[**MultiStepForecasting**]

Пакет *sktime*[**sktime**] предоставляет нам эти функциональные возможности с удобным API. Мы используем метод $create\_forecaster()$, в котором функция $make\_reduction()$ оборачивает $LGBMRegressor()$ модель и преобразует входные временные ряды в табличный формат, когда мы тренируем нашу прогнозную модель.

Мы также используем метод $ForecastingGridSearchCV()$ для поиска оптимальной длины окна ($window\_length$) лаг признаков. [**LightgbmSktime**]

Для выполнения задачи прогнозирования временных рядов в данной работе будет использоваться набор данных «Sunsposts» [**Sunspots**], который является набором наблюдений за возникновением пятен на Солнце (среднемесячное количество пятен на Солнце).

1.4. Вычислительный эксперимент и анализ ошибки

Ссылка на код работы

После загрузки данных из набора «Sunsposts», данные были визуализированы с помощью средств языка Python (библиотека *matplotlib*). График временного ряда показан рис. 1.4. [**tsplot**]

Перед непосредственным прогнозированием, данные должны быть проанализированы на стационарность. [**Stationarity**] Стационарность означает, что временной ряд не меняет своих статистических свойств с течением времени, в частности, своего среднего значения и дисперсии. Временные ряды с циклическим поведением в основном стационарны, в то время как временные ряды с тенденциями или сезонностью не являются стационарными.

Для проверки стационарности, мы можем использовать графики автокорреляции (ACF) и частичной автокорреляции (PACF), а также выполняет дополненный тест Дики-Фуллера.[**DickeyFuller**] Результаты проверки показаны на рис. 1.5.

График автокорреляции (ACF) может быть использован для определения стационарности временного ряда. График частичной автокорреляции (PACF) полезен для определения порядка авторегрессии. Дополненный единичный тест Дики-Фуллера проверяет, является ли временной ряд нестационарным. Нулевая гипотеза заключается в том, что ряд нестационарен, следовательно, если p-значение

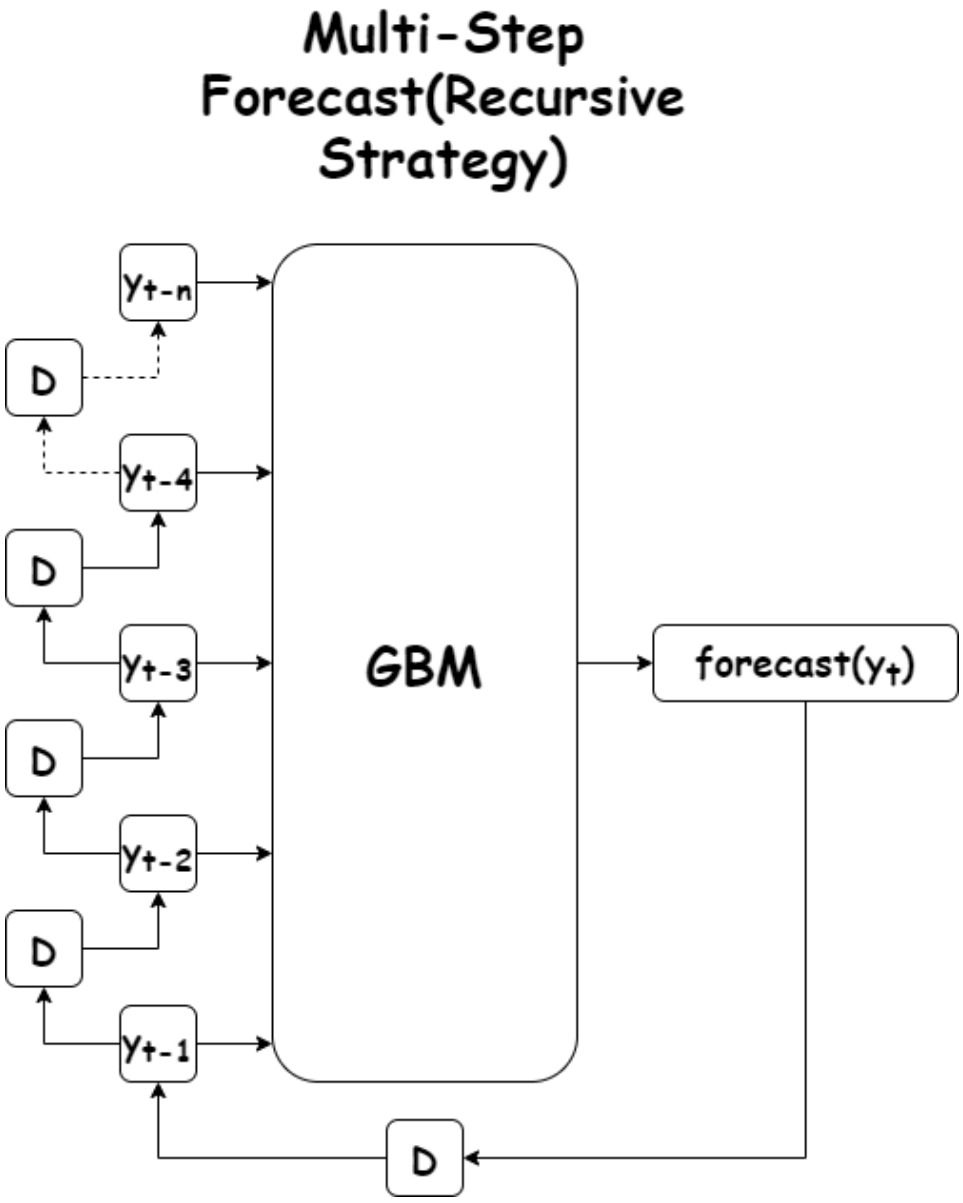

Рис. 1.3. "Multi Step Forecasting"

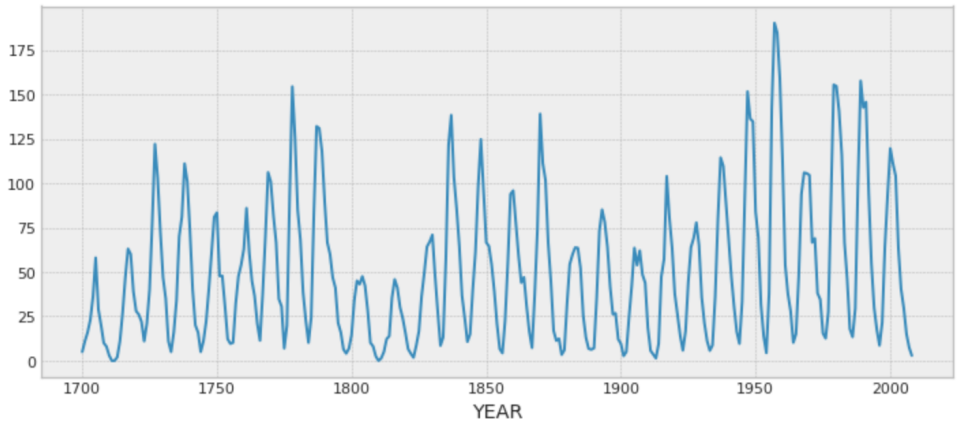

Рис. 1.4. "Набор данных Sunspots"

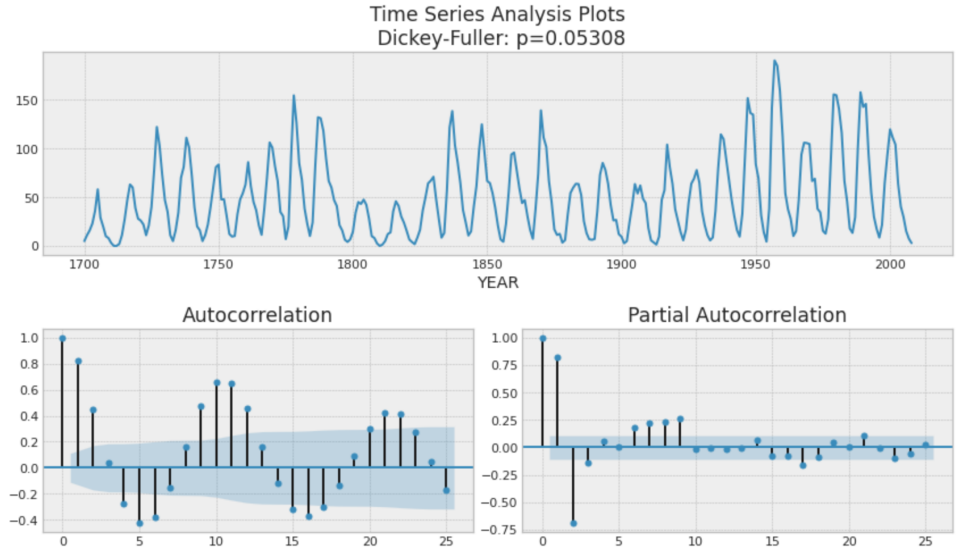

Рис. 1.5. "Анализ временных рядов"

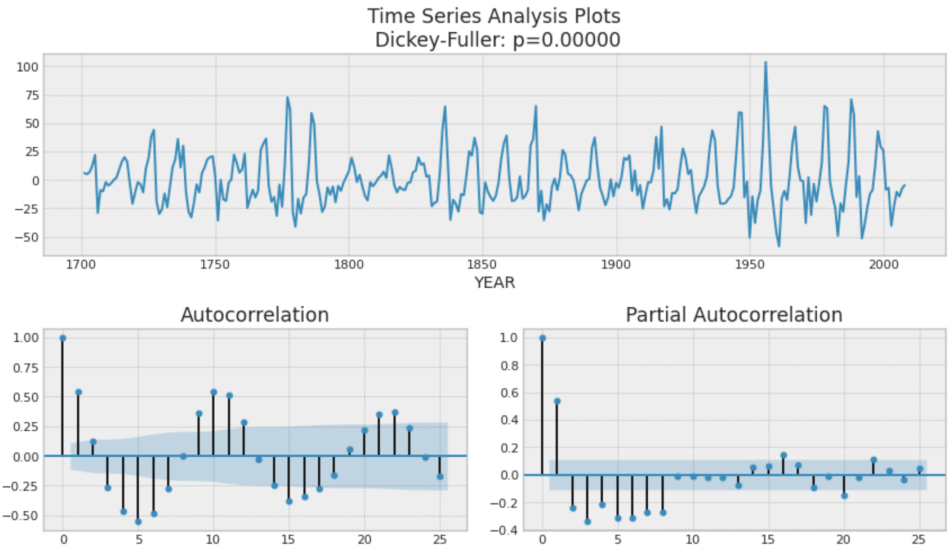

Рис. 1.6. "Анализ лага временных рядов"

мало, это означает, что временной ряд стационарен.

На рисунке выше p-значение теста Дики-Фуллера недостаточно значимо. Первую разность (лаг) будет использована, чтобы сделать ряд более стационарным (см. рис. 1.6). Для анализа ошибки предсказания были использованы классические метрики MAE (средняя абсолютная ошибка) и MAPE (средняя абсолютная процентная ошибка):

$$MAE = \frac{\sum_{t=1}^{n} |y_t - \hat{y}_t|}{n}$$

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i}$$

Протренировав LightGBM на трансформированной выборке с лагом 1, были протестированы различные значения параметра $window\_length$ (длина окна предсказания) для функции $make\_reduction$, который позволяет в зависимости от величины параметра формировать тренировочную выборку и строить на ней прогноз. Лучший результат был достигнут при $window\_length = 25$: MAE = 15.66, MAPE=1.58 (см. рис. 1.7).

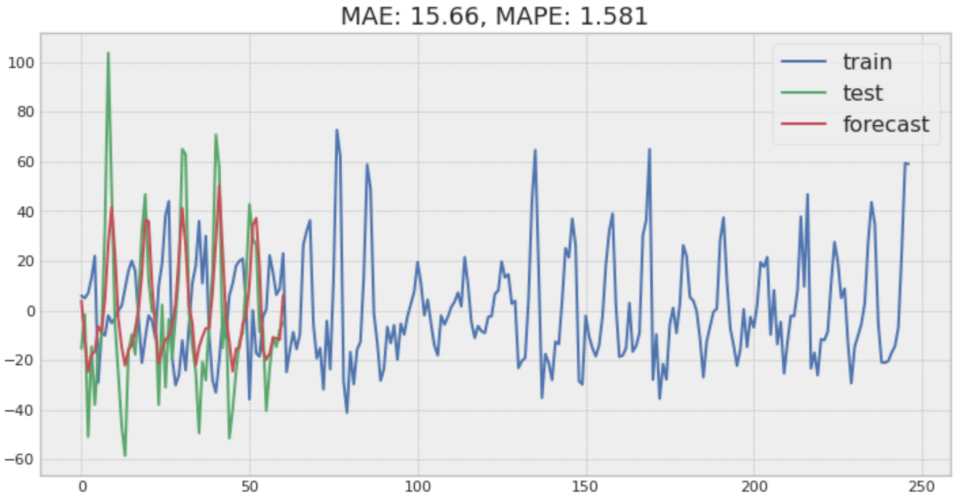

Рис. 1.7. "Результаты ошибки на тестовой выборке"

window	**MAE**	**MAPE**
5	27.56	2.54
10	18.19	2.59
15	21.76	3.35
20	20.36	2.96
30	16.08	1.69

Таблица 1.1. Результаты ошибок MAE и MAPE для других значений параметра window length