



ИЗКУСТВЕН ИНТЕЛЕКТ

Реферат на тема „Градиентно усилване“

Явор Йорданов Чамов | СИТ 1 А) | №21621577

Съдържание

Въведение	3
Какво е <i>градиент</i> ?	3
Какво е <i>Boosting</i> алгоритъм?	4
Какво представява Gradient Boosting?	5
Исторически сведения	6
Ключови концепции (понятия) при работа с GBM.....	6
Тълкуване на алгоритъма	7
Планиране на алгоритъма чрез псевдокод.....	9
Регуляризация	9
Обобщени предимства	10
Предизвикателства и ограничения.....	11
Популярни варианти на алгоритъма.....	11
Употреба	12
Бъдещето на методите за усилване на градиента.....	12
Заключение	13
Ресурси.....	13

Въведение

Машинното обучение промени начина, по който се решават сложни проблеми в различните индустрии. Сред различните налични алгоритми, *Gradient Boosting Machines (GBM)* се очертава като една от най-мощните техники за прогнозно моделиране, особено в структурирани/таблични данни. *GBM* принадлежи към методите за обучение в група, които имат за цел да подобрят точността на прогнозиране чрез комбиниране на множество модели. Уникалният аспект на *GBM* се крие в неговия последователен подход, при който моделите се обучават итеративно, за да коригират грешките на предишни модели. Този реферат разглежда основите на *GBM*, неговите принципи на работа, предимства, ограничения и приложения в различни области.

Какво е градиент?

В контекста на градиентното усилване, **градиентът** представлява частна производна на функцията на загуба спрямо предсказанията на модела. Той определя **количествено колко би се променила загубата**, ако предсказанията на модела бъдат коригирани. Градиентът се използва за насочване на процеса на обучение, като показва посоката и величината на необходимите подобрения за минимизиране на функцията на загуба.

Частната производна е мярка за това как една функция се променя при промяна на една от нейните променливи, като същевременно всички останали променливи остават постоянни. Това е основна концепция в многопроменливото смятане и се използва за анализиране на поведението на функции, които зависят от повече от една променлива.

Функцията на загуба $L(y, \hat{y})$ измерва **разликата между истинските стойности y и предвидените стойности \hat{y}** от модела.

- Пример за регресия: Mean Squared Error (MSE)

При всяка итерация на *gradient boosting*, алгоритъмът изчислява градиента на функцията на загуба по отношение на текущите прогнози:

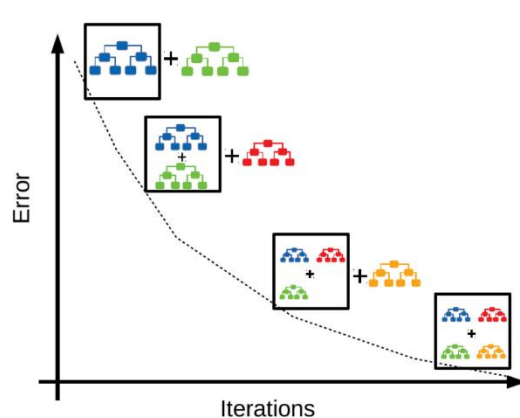
$$g_i = \frac{\partial L(y_i, \hat{y}_t)}{\partial \hat{y}_t}$$

Този градиент g_i представлява **посоката и големината на корекцията**, необходима за прогнозиране на всяка точка от данни i .

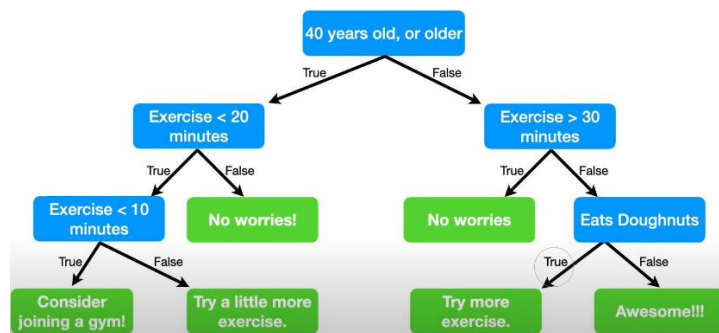
Градиентът действа като "сигнал за грешка" или "коригиращ сигнал". Той казва на модела как да актуализира своите прогнози, за да намали функцията на загуба.

При метода за усилване на градиента, отрицателният градиент $-g_i$ се използва като цел за следващия слаб обучаем (напр. дърво на решенията). Слабият учещ се обучава да приближава този отрицателен градиент. Градиентът при *Gradient Boosting* представлява посоката на най-стръмното изкачване на функцията на загубата. За да се минимизира функцията на загубата (което е целта на оптимизацията), моделът трябва да се движи в

обратната посока - тоест посоката на най-стръмното спускане. Ето защо отрицателният градиент се използва като цел за следващия слаб ученик.



Дървото на решенията е предсказващ модел, който използва дървовидна структура за представяне на решения и техните възможни последствия, включително резултати, разходи или вероятности. Обикновено се използва като „слаб обучаем“ в алгоритми за усиление на градиент като $h_{\mathcal{M}}(x)$, където се използват плитки (с малка дълбочина) дървета на решения за итеративно подобряване на цялостния модел.



Какво е *Boosting* алгоритъм?

Алгоритъмът „усиление“ попада в категорията на груповите методи. Това означава, че вместо да обучаваме един силен модел, ние обучаваме много „прости“, които заедно дават по-добри резултати от единичен силен. „Усливането“ работи чрез последователно обучение на множество прости модели. Всеки следващ модел се фокусира върху грешките, направени от предходните модели. Като резултат прогнозите се основават на всички създадени модели.

BOOSTING



Какво представлява Gradient Boosting?

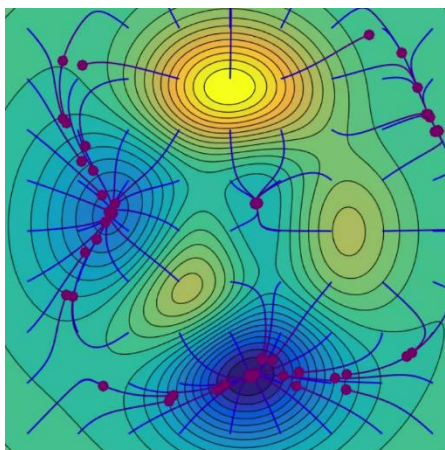
Gradient boosting е контролирана (*supervised*) техника за машинно обучение, предназначена за задачи от тип регресия и класификация. Разчита се на група от слабо обучаеми, обикновено дървета на решенията, за да се изгради силен прогнозен модел. Терминът „усилване на градиента“ идва от начина, по който алгоритъмът оптимизира модела. Итеративно **минимизира функцията на загуба** чрез **градиентно спускане**.

*Градиентно спускане е метод за намиране на **локален минимум** на диференцируема функция. Идеята е чрез итеративен подход да се намери най-ниските стойности на функцията, като се правят постепенни стъпки в посока обратна на градиента.*

$$\theta_{new} = \theta_{old} - \eta \cdot \nabla L(\theta)$$

$\nabla L(\theta)$: Градиентът на функцията на загубата.

η : Скоростта на обучение, контролираща размера на стъпката.



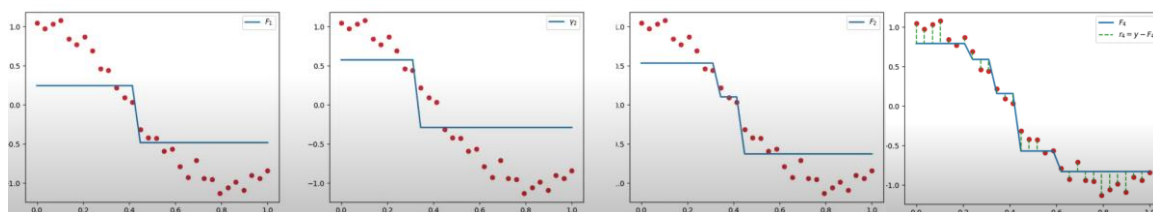
Основната идея остава изграждането на много слаби (*weak*) учащи, точно както при стандартния *boosting*. Този път, в случай на усилване на градиента, всеки следващ модел не се фокусира директно върху грешките на предишния, а върху така наречените *псевдоостатъци*.

Псевдоостатъците са количествата, използвани за **измерване на грешката** или несъответствието на текущия модел на групата при всяка итерация. Те се наричат псевдо, защото не са истински остатъци в класическия смисъл (разлики между наблюдаваните и прогнозираните стойности), а по-скоро градиенти на функцията на загуба по отношение на прогнозите на текущата група.

С прости думи, обикновеното усилване (само *boosting*) се фокусира върху класическите остатъци, които са разликите между истинските стойности и прогнозираните. Следователно, той точно научава тези остатъци. Градиентното усилване, от друга страна, има по-широка перспектива. Той не се фокусира директно върху остатъците (разликите), а по-скоро върху посоката, в която функцията на загуба приема минималната стойност.

Характеристика	Boosting	Gradient Boosting
Фокус	Класически остатъци	Минимизиране на функцията на загубите чрез използване на градиента
Обучение	Адаптиране въз основа на грешната класификация	Използва градиента на функцията на загубите за актуализации

За разлика от други методи на групиране като *bagging* или *random forests*, които обучават модели независимо и ги комбинират, *gradient boosting* работи последователно. Всеки модел в последователността е обучен да коригира грешките на своя предшественик. Това итеративно подобряване прави метода много ефективен за улавяне на сложни връзки в данните.



Исторически сведения

Основата за усилване на градиента е положена от Лео Брейман, който въвежда концепцията за „усилване“ през 90-те години. Техниката по-късно е усъвършенствана от Джеръм Фридман, който въвежда идеята за оптимизация, базирана на градиент, което довежда до разработването на модерни *GBM* (*Gradient Boosting Machines*).

Ключови концепции (понятия) при работа с GBM

За да се разбере как работи *GB*, е важно да се обяснят следните концепции:

1. Ансамбловото обучение:

- Ансамбловите методи комбинират множество модели, за да подобрят общото представяне. В градиентните бустинг машини (*GBM*), ансамбълът се състои от слаби обучаеми, обикновено плитки дървета на решенията, които се тренират последователно.

2. Слаби обучаеми:

- „Слабият обучаем“ е модел, който се представя малко по-добре от този на случайното предположение (*random guessing*). Дървета на решенията с плитка дълбочина (например, 3-5 нива) често се използват в реализациите.

3. Градиентно спускане:

- Оптимизира модела чрез минимизиране на функцията на загуба. Функцията на загуба измерва разликата между предсказаните и действителните стойности.

4. Функция на загуба:

- Изборът на функция на загуба зависи от задачата:

- Регресия: Средна квадратична грешка (MSE).
- Класификация: Логаритмична загуба или ентропия.

5. Скорост на обучение:

- Скоростта на обучение контролира приноса на всеки слаб обучаем към крайния модел. По-ниска скорост на обучение често подобрява точността, но изисква повече итерации.

6. Регуляризация:

- Методът за усилване на градиента включва техники за регуляризация (например свиване, контрол на дълбочината на дървото), за да предотврати прекомерното пренастройване (*overfitting*).

Пренастройването при усилване на градиента възниква, когато моделът срещне поведение, шум или произволни колебания в данните за обучение, които не се обобщават до реални (невидими) данни. Това води до отлично представяне на обучителния набор от данни, но лошо представяне на тестови или валидиращи данни, което показва, че моделът е станал твърде специфичен за обучаващия набор.

Тълкуване на алгоритъма

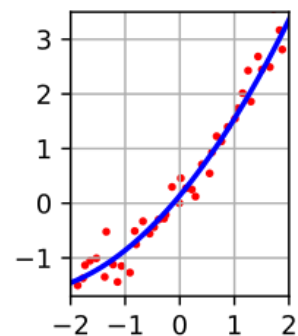
Подобно на други методи за усилване, градиентното усилване итеративно комбинира слаби („обучаеми“) в един силен „обучен“ модел. Най-лесно е да се обясни в контекста на регресията с най-малки квадрати, където целта е да се "научи" модел (F) да предсказва стойности от вида $\hat{y} = F(x)$ чрез минимизиране на средната квадратична грешка $\frac{1}{n} \sum_i (\hat{y}_i - y_i)^2$, където (i) индексира някакъв набор за обучение с размер (n) от действителни стойности на променливата на изхода (y) .

\hat{y}_i = предвидената стойност $F(x_i)$

y_i = наблюдаваната стойност

n = броят на пробите в y

Методът на най-малките квадрати е метод за оценка на параметрите в регресионния анализ, базиран на минимизиране на сумата от квадратите на остатъците. Остатъкът е разликата между наблюдавана стойност и стойността, предвидена от модела в резултатите на всяко отделно уравнение. По-просто казано, методът на най-малките квадрати е математическа процедура за намиране на най-подходящата крива за даден набор от точки чрез минимизиране на сумата от квадратите на отклоненията ("остатъците") на точките от кривата.



Средната квадратна грешка (MSE) е често използвана функция за загуба при регресионни проблеми. Тя измерва средната квадратна разлика между действителните (истински) стойности и прогнозираните стойности. Определя количествено колко добре регресионният модел прогнозира целевата променлива.

Ако алгоритъмът има M етапа, на всеки етап m ($1 \leq m \leq M$), да предположим някакъв несъвършен модел F_m (за ниско m , този модел може просто да предвиди \hat{y}_i да бъде \bar{y} , средната стойност на y). За да подобрим F_m , нашият алгоритъм трябва да добави нов основен оценител (слаб обучаем), $h_m(x)$.

В контекста на градиентното усилване, $h_m(x)$ е слаб обучаем или основен оценител, който се обучава на m -тия етап на алгоритъма. Неговата роля е да коригира грешките, направени от съществуващия модел $F_m(x)$ и да подобри цялостната прогноза.

На всеки етап m , слабият обучаем $h_m(x)$ се обучава да приближава остатъците, които представляват грешките, направени от текущия модел $F_m(x)$.

Остатъците се изчисляват като:

$$r_i = y_i - F_m(x_i)$$

$h_m(x)$ научава разликата между истинските целеви стойности y_i и прогнозите на текущия модел $F_m(x_i)$.

След обучение на $h_m(x)$, моделът $F_m(x)$ се актуализира чрез добавяне на $h_m(x)$, мащабиран със скорост на обучение η :

$$F_{m+1}(x) = F_m(x) + \eta \cdot h_m(x)$$

$$F_{m+1}(x_i) = F_m(x_i) + h_m(x_i) = y_i - F_m(x_i) + F_m(x_i) = y_i$$

Следователно градиентното усилване ще приспособи h_m към остатъка $y_i - F_m(x_i)$. Както при други варианти на усилване, всеки F_{m+1} се опитва да коригира грешките на своя предшественик F_m .

Наблюдава се, че остатъците $h_m(x_i)$ за даден модел са пропорционални на отрицателните градиенти на функцията на загуба на средната квадратна грешка (MSE) по отношение на $F(x_i)$.

$$L_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (y_i - F(x_i))^2$$
$$-\frac{\partial L_{\text{MSE}}}{\partial F(x_i)} = \frac{2}{n} (y_i - F(x_i)) = \frac{2}{n} h_m(x_i)$$

Gradient boosting може да се разглежда като рамка, която обхваща алгоритъма на градиентно спускане. Основната идея е, че градиентното усилване итеративно минимизира определена функция на загуба чрез изграждане на ансамбъл от слаби „обучаеми“ (напр. дървета на решения), ръководени от градиента на функцията на загуба на всяка стъпка.

Планиране на алгоритъма чрез псевдокод

Вход: множество за обучение $\{ (x_i, y_i) \mid i = 1, \dots, n \}$, диференцируема функция на загубата $L(y, F(x))$, брой итерации M .

Алгоритъм:

1. Инициализиране на модела с постоянна стойност:

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

Инициализира модела $F_0(x)$, който е постоянната прогноза, която минимизира общата загуба L за всички точки от данни.

2. For $m = 1$ to M :

- a. Изчисляване на така наречените псевдоостатъци:

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \dots, n.$$

- b. Поставяне на основен обучаем (или слаб обучаем, напр. дърво) $h_m(x)$ към псевдоостатъците
- c. Изчисляване на множител γ_m чрез решаване на следния едномерен проблем за оптимизация:

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

Определяне на оптималния размер на стъпката (коефициент на мащабиране) γ_m в m -тия етап на алгоритъма. Кorigира се приноса на слабия обучаем $h_m(x)$, за да минимизира функцията на загуба L .

- d. Актуализиране на модела:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

3. Извеждане на $F_M(x)$.

Регуляризация

Регуляризацията в *Gradient boosting* се отнася до техники, използвани за предотвратяване на пренастройване и подобряване на способността за генерализиране на модела. *Gradient boosting* е мощен алгоритъм, но е склонен към пренастройване, тъй като може да продължи да подобрява точността на обучението, като напасва шума в данните. Методите за регуляризация контролират това поведение и помагат на модела да работи добре върху неясни данни.

По-долу са представени често използвани начини за постигане на регуляризация:

1. Скорост на обучение (свиване)

- Скоростта на учене η мащабира приноса на всеки слаб обучаем (напр. дърво на решенията) към общия модел

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x)$$
 - По-малък η намалява въздействието на всеки слаб обучаем, което прави актуализирането на модела по-предпазливо
 - Ефект:
 - По-малката скорост на обучение изисква повече итерации (или дървета), но води до по-добра генерализация
 - Балансира размера на стъпката на актуализациите, за да избегне прекомерното пренастройване
2. Дървовидно регулиране
- Регуларизация може да се приложи и към отделните дървета на решенията, използвани като слаби обучаеми
 - Максимална дълбочина
 - Ограничава дълбочината на всяко дърво на решенията, за да контролира неговата сложност
 - Плитките дървета намаляват риска от прекомерно пренастройване, като избягват прекалено специфични разцепвания
 - Минимални пробни листа
 - Гарантира, че минимален брой проби трябва да присъстват в листен възел
 - Предотвратява разпределение, което е твърде фино
 - Максимален брой листа
 - Ограничава общия брой листни възли в едно дърво, за да ограничи размера му
 - Характеристики на подизвадка
 - Използване на подмножество от белези за разделяне на всеки възел, намалявайки дисперсията
3. Подсемплиране (усилване на стохастичен градиент)
- Вместо да се използва целият набор от данни за обучение на всеки слаб обучаем, се взема произволна подгрупа от данни без подмяна
 - Ефект:
 - Въвежда произволност, която намалява пренастройването и увеличава здравината на модела.

Обобщени предимства

Gradient boosting предлага няколко предимства, които го правят предпочитан избор за специалистите по машинно обучение:

- Висока предсказателна мощност:**
 - Често постига резултати от най-високо ниво в състезания като Kaggle (*Онлайн платформа и общност за ентусиасти в областта на науката за данни и машинното обучение за сътрудничество, състезание и споделяне на знания. Той е широко признат като една от най-популярните платформи за предизвикателства и проекти в областта на науката за данни*).
- Гъвкавост:**
 - Поддържа различни функции на загуба, което го прави подходящ за разнообразни задачи.
- Важност на характеристиките:**

- Предоставя информация за това кои характеристики допринасят най-много за прогнозираните резултати.
- 4. **Работа с нелинейни зависимости:**
 - Последователният му характер позволява да улавя сложни модели в данните.
- 5. **Устойчивост на изключително големи отклонения:**
 - По-малко чувствителен към отклонения в данните в сравнение с някои други алгоритми.

Предизвикателства и ограничения

Въпреки предимствата си, *Gradient Boosting* има някои ограничения:

1. **Интензивност на изчисленията:**
 - Обучението може да бъде бавно, особено при големи набори от данни.
2. **Настройка на хиперпараметри:**
 - Изисква внимателно настройване на хиперпараметри като скорост на обучение (*learning rate*) и дълбочина на дървото.
3. **Пренастройване (*Overfitting*):**
 - Без подходяща регуляризация може да се пренасити към тренировъчните данни.
4. **Използване на памет:**
 - Алгоритъмът може да бъде паметно-интензивен, особено при работа с големи набори от данни.

Популярни Варианти на алгоритъма

Разработени са няколко оптимизирани реализации, за да се преодолеят част от ограниченията:

Алгоритъм	Описание
XGBoost	Известен със своята мащабируемост и ефективност, <u><i>XGBoost</i></u> включва функции като паралелна обработка и регуляризация.
LightGBM	Фокусиран върху скоростта, <i>LightGBM</i> използва подход, базиран на хистограма, за по-бързо обучение.
CatBoost	Създаден за категорийни данни, <i>CatBoost</i> автоматизира обработката на категорийни променливи.

XGBoost (*eXtreme Gradient Boosting*) е високоефективна и мащабируема реализация на алгоритъма за градиентно усилване. Широко използван е в машинното обучение за структурирани (таблични) данни. Проектиран е да оптимизира както изчислителната скорост, така и производителността на модела, което го прави един от най-популярните инструменти за задачи за прогнозно моделиране. *XGBoost* използва усилване на

градиента като свой основен алгоритъм, където слабите обучаеми (обикновено дървета на решения) се добавят последователно, за да се коригират грешките, направени от предишни модели. Включва L1 (*Lasso*) и L2 (*Ridge*) регулация в обективната (математическа формулировка, която моделът има за цел да минимизира) си функция за предотвратяване на пренастройването и подобряване на генерализацията.

$$L_{regularized} = \sum L(y_i, \hat{y}_i) + \lambda \cdot \text{complexity of the tree}$$

XGBoost използва многонишкова обработка за ефективно разрастване на дървета, което го прави значително по-бърз от традиционните методи за повишаване на градиента. Използва параметър "максимална дълбочина" и извършва подрязване по време на изграждане на дърво, осигурявайки изчислителна ефективност чрез елиминиране на ненужните разделяния. Обработва ефективно липсващи или оскъдни данни, като използва указания по подразбиране за липсващи стойности. Позволява на потребителите да дефинират персонализирани функции за загуба, за да отговарят на конкретни задачи. Осигурява оценки за важност на характеристиките, като помага да се определи кои променливи са най-влиятелни при правенето на прогнози.

Употреба

Градиентното усилване може да се използва в областта на обучението за класиране. Уеб търсачки като *Yahoo* и *Yandex* използват варианти на градиентно усилване в своите машинно обучени модели за класиране на резултати. Увеличаването на градиента също се използва във физиката на високите енергии при анализ на данни. Например, в Големия адронен колайдер (*LHC*), градиентно усилващите дълбоки невронни мрежи (*DNN*) успешно репликираха резултатите от традиционните методи за немашинно обучение върху набори от данни, използвани за откриването на бозона на Хигс. Дървото на решенията с градиентно усилване също се прилага при земни и геоложки проучвания.

Област	Приложение
Финанси	Откриване на измами и моделиране на кредитен риск.
Здравеопазване	Прогноза на заболявания и анализ на повторно приемане на пациенти.
Маркетинг	Сегментация на клиенти и препоръчителни системи.
Електронна търговия	Прогнозиране на търсенето и динамично ценообразуване.

Бъдещето на методите за усилване на градиента

С напредъка в хардуера и софтуера, методите стават по-бързи и по-ефективни. Интегрирането им с техники за дълбоко обучение, както се вижда в хибридните модели, има голям потенциал за бъдещи приложения.

Заклучение

Gradient boosting е основен елемент на съвременното машинно обучение. Неговата способност да обработва сложни набори от данни и да предоставя точни прогнози го прави незаменим в различни индустрии. Въпреки че изисква внимателно настройване и изчислителни ресурси, резултатите, които предоставя, оправдават вложените усилия. С развитието на областта на машинното обучение, вероятно ще остане ключов инструмент за решаване на сложни проблеми.

Ресурси

Gradient boosting. (2024, October 2). In Wikipedia.
https://en.wikipedia.org/wiki/Gradient_boosting

Boosting (machine learning). (2024, November 13). In Wikipedia.
[https://en.wikipedia.org/wiki/Boosting_\(machine_learning\)](https://en.wikipedia.org/wiki/Boosting_(machine_learning))

Errors and residuals. (2023, November 25). In Wikipedia.
https://en.wikipedia.org/wiki/Errors_and_residuals

Decision tree learning. (2024, July 16). In Wikipedia.
https://en.wikipedia.org/wiki/Decision_tree_learning

Loss function. (2024, November 27). In Wikipedia.
https://en.wikipedia.org/wiki/Loss_function

Least squares. (2024, October 16). In Wikipedia. https://en.wikipedia.org/wiki/Least_squares

Mean squared error. (2024, June 11). In Wikipedia.
https://en.wikipedia.org/wiki/Mean_squared_error

Gradient descent. (2024, November 14). In Wikipedia.
https://en.wikipedia.org/wiki/Gradient_descent

Gradient Boosting Machine Explained in Detail. (2024, November 12). In Medium.
<https://medium.com/@msoczi/gradient-boosting-machine-explained-in-detail-b3d24aaecfd4>

Gradient Boosting with Regression Trees Explained. (2023, February 24). In YouTube.
<https://youtu.be/lOwsMpdjxog?si=nwmvpTGWkCHiDFYC>