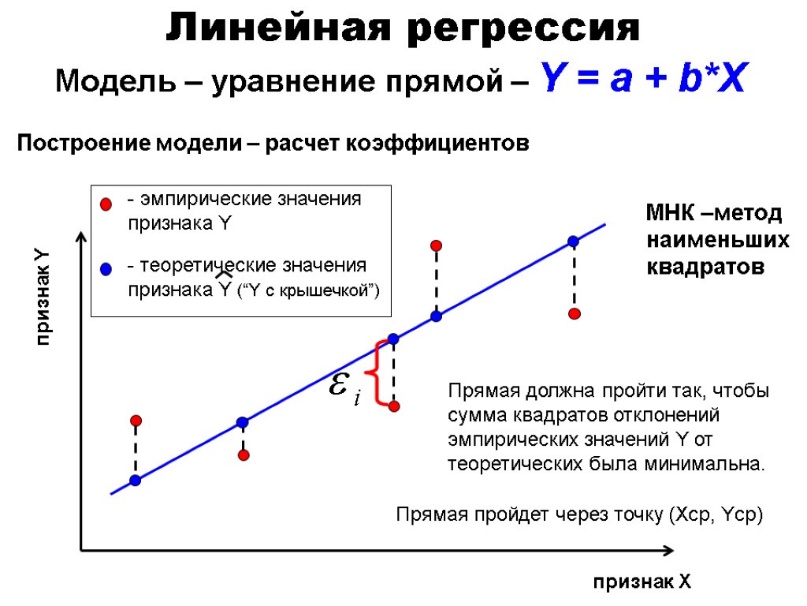
Рубежный контроль 1

по курсу «Методы машинного обучения»

Вопросы:

1. Линейная регрессия. Назначение, формула, функция потерь, способы обучения (без вывода)

Линейная регрессия — это статистический метод, который используется для моделирования и анализа взаимоотношений между зависимой переменной (целевой переменной) и одной или несколькими независимыми переменными (факторами). Она применяется для предсказания значения зависимой переменной на основе значений независимых переменных.



Назначение линейной регрессии:

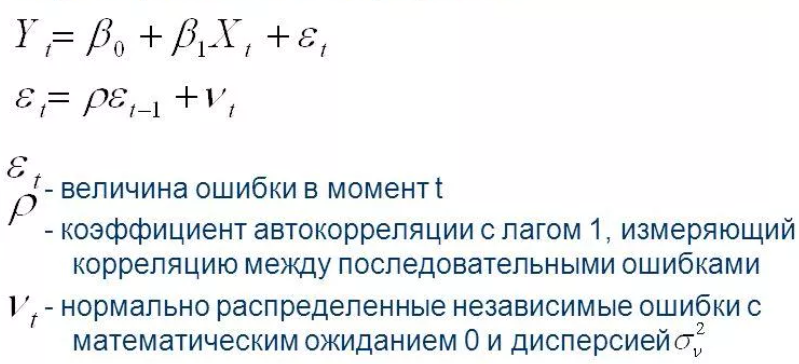
1. \*\*Предсказание\*\*: Определение значений целевой переменной на основе новых значений независимых переменных.

2. \*\*Анализ зависимости\*\*: Оценка силы и характера зависимости между переменными.

3. \*\*Интерпретация коэффициентов\*\*: Понимание влияния каждой независимой переменной на зависимую.

Формула линейной регрессии

Общая формула линейной регрессии с одной независимой переменной:



- \( \beta\_0 \) — интерсепт (свободный член),

- \( \beta\_1 \) — коэффициент наклона (показывает изменение \( y \) при изменении \( x \) на единицу),

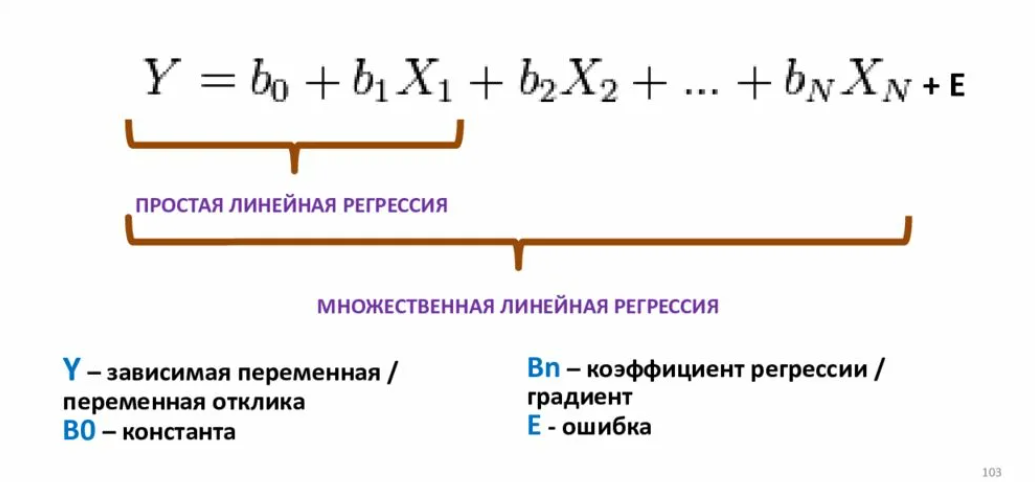
- \( y \) — зависимая переменная (то, что мы пытаемся предсказать),

- \( x \) — независимая переменная (фактор),

- \( \beta\_0 \) — интерсепт (свободный член),

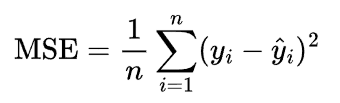
- \( \varepsilon \) — случайная ошибка (остаток), отражающая отклонения предсказаний от фактических значений.

Для множественной линейной регрессии:



Функция потерь

Наиболее часто используемая функция потерь для линейной регрессии — это среднеквадратичная ошибка (MSE, Mean Squared Error):



- \( n \) — количество наблюдений,

- \( y\_i \) — фактические значения,

- \( \hat{y}\_i \) — предсказанные значения.

Способы обучения

Основная задача обучения линейной регрессии — это найти такие значения коэффициентов  , которые минимизируют функцию потерь. Существует несколько методов для этого:

1. \*\*Метод наименьших квадратов (OLS, Ordinary Least Squares)\*\*:

- Наиболее распространенный метод. Он ищет параметры, минимизирующие сумму квадратов ошибок (разниц между предсказанными и фактическими значениями).

- Решение находится аналитически, путем решения системы линейных уравнений, что обеспечивает точное решение.

2. \*\*Градиентный спуск\*\*:

- Итеративный метод, который использует производные функции потерь для обновления коэффициентов в направлении, уменьшающем ошибку.

- Подходит для больших наборов данных и в случаях, когда аналитическое решение сложно получить.

3. \*\*Регуляризация\*\*:

- Включает методы, которые добавляют штраф за большие значения коэффициентов, чтобы избежать переобучения.

- \*\*Ридж-регрессия (L2-регуляризация)\*\*: добавляет штраф за квадрат нормы коэффициентов.

- \*\*Лассо-регрессия (L1-регуляризация)\*\*: добавляет штраф за абсолютную величину коэффициентов, что может приводить к нулевым коэффициентам для некоторых переменных, осуществляя тем самым отбор признаков.

4. \*\*Метод стохастического градиентного спуска (SGD, Stochastic Gradient Descent)\*\*:

- Вариант градиентного спуска, который обновляет параметры после обработки каждого отдельного обучающего примера, что делает его более подходящим для очень больших наборов данных.

1. Оценка параметров линейной регрессии посредством градиентного спуска

Градиентный спуск — это метод, используемый для нахождения минимального значения функции. В контексте линейной регрессии, мы используем его, чтобы минимизировать функцию потерь, которая показывает, насколько наши предсказания отличаются от реальных значений.

### Принцип работы

Представьте, что вы стоите на вершине холма и хотите спуститься в самую низкую точку. Вы будете делать маленькие шаги вниз по склону, каждый раз проверяя, в каком направлении наклон круче всего. Таким образом, вы постепенно спускаетесь вниз, пока не достигнете самой низкой точки.

### Применение в линейной регрессии

В линейной регрессии наша цель — найти такие значения параметров (коэффициентов) \( \beta\_0 \) и \( \beta\_1 \), которые минимизируют функцию потерь (например, среднеквадратичную ошибку, MSE).

#### Шаги градиентного спуска

1. \*\*Инициализация параметров\*\*:

- Начинаем с произвольных значений параметров и (чаще всего они устанавливаются в нули или случайные небольшие числа).

2. \*\*Вычисление функции потерь\*\*:

- Для каждой точки данных вычисляем, насколько наши предсказания \( \) отличаются от реальных значений \( y \). Это дает нам общую ошибку (MSE).

3. \*\*Вычисление градиента\*\*:

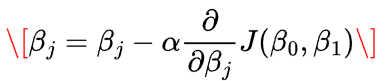
- Градиент — это вектор, указывающий направление и скорость, с которой нужно изменять параметры, чтобы уменьшить ошибку.

- Для простоты, можно представить градиент как наклон функции потерь относительно каждого параметра.

4. \*\*Обновление параметров\*\*:

- Обновляем параметры в направлении, противоположном градиенту, чтобы уменьшить ошибку.

- Обновление происходит по формуле:



где \( \alpha \) — это скорость обучения (learning rate), определяющая размер шага, который мы делаем в направлении градиента.

5. \*\*Повторение\*\*:

- Повторяем шаги 2-4, пока изменения в функции потерь не станут незначительными, что означает, что мы приблизились к минимальному значению функции потерь.

Пример

Представьте, что у вас есть набор данных с количеством часов, потраченных на учебу (независимая переменная \( x \)) и оценками студентов (зависимая переменная \( y \)). Мы хотим найти такие параметры \( \beta\_0 \) и \( \beta\_1 \), чтобы модель \( y = \beta\_0 + \beta\_1 x \) наилучшим образом предсказывала оценки.

1. \*\*Инициализируем параметры\*\*: пусть \( \beta\_0 = 0 \) и \( \beta\_1 = 0 \).

2. \*\*Вычисляем предсказания и ошибку\*\*: допустим, первые предсказания сильно отличаются от реальных оценок.

3. \*\*Вычисляем градиент\*\*: определяем, в каком направлении и на сколько нужно изменить \( \beta\_0 \) и \( \beta\_1 \).

4. \*\*Обновляем параметры\*\*: корректируем \( \beta\_0 \) и \( \beta\_1 \) по формуле, например, \( \beta\_0 \) и \( \beta\_1 \) увеличиваются или уменьшаются.

5. \*\*Повторяем\*\*: снова вычисляем предсказания, ошибку и обновляем параметры, постепенно приближаясь к минимальной ошибке.

Таким образом, с каждым шагом наши параметры \( \beta\_0 \) и \( \beta\_1 \) становятся всё точнее, и модель всё лучше предсказывает оценки студентов на основе количества часов, потраченных на учебу.

1. Ошибка обучения, ошибка тестирования, переобучение. Привести пример в виде графика с кривыми ошибок

вапрол

1. Полиномиальная регрессия. Назначение, формула, функция потерь, способы обучения (без вывода)
2. Логистическая регрессия. Назначение, формула, функция потерь, способы обучения (без вывода)

Логистическая регрессия — это метод машинного обучения, используемый для классификации, то есть для предсказания категорий или классов. В отличие от линейной регрессии, которая предсказывает непрерывные значения, логистическая регрессия предсказывает вероятность того, что наблюдение принадлежит к определенному классу.

### Назначение логистической регрессии

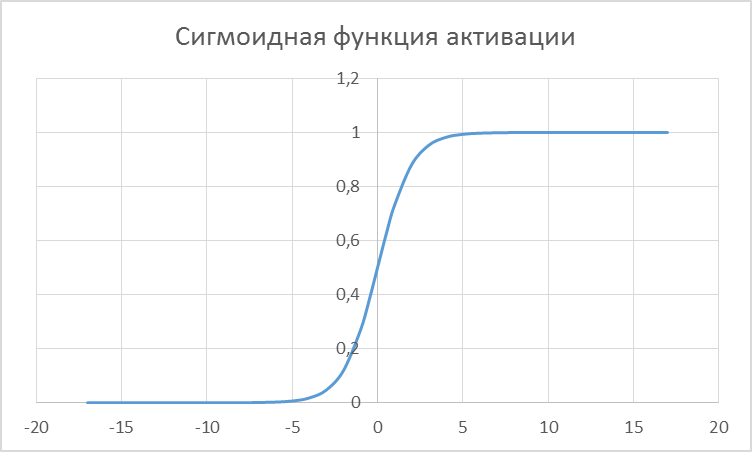
Основная цель логистической регрессии — предсказывать вероятность принадлежности наблюдения к одному из двух классов (бинарная классификация). Например, она может использоваться для:

- Предсказания, будет ли клиент покупать продукт (да/нет).

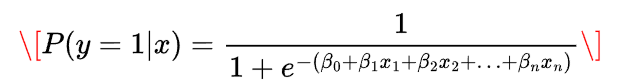
- Определения, является ли электронное письмо спамом (да/нет).

- Диагностики, болен ли пациент (да/нет).

### Формула логистической регрессии



Вместо прямой линии, как в линейной регрессии, логистическая регрессия использует логистическую функцию (сигмоидную функцию) для преобразования линейного предсказания в вероятность. Формула выглядит так:



Где:

- P(y=1 | x) — вероятность того, что целевая переменная y равна 1 при данном значении переменных x .

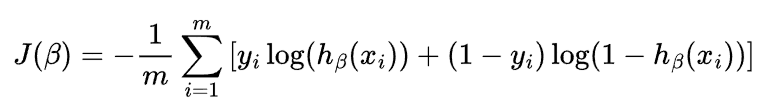
-  — параметры модели.

-  — независимые переменные (факторы).

### Функция потерь

Кросс-энтропия – это мера различия между двумя вероятностными распределениями.

Для логистической регрессии используется функция потерь, называемая логистической (кросс-энтропийной) функцией потерь. Она измеряет, насколько хорошо предсказанные вероятности соответствуют фактическим меткам классов. Формула выглядит так:



Где:

- m — количество наблюдений.

-  — фактическое значение целевой переменной (0 или 1) для i-го наблюдения.

- — предсказанная вероятность для i-го наблюдения.

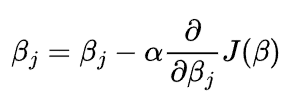
### Способы обучения

Как и в линейной регрессии, наша цель — найти такие значения параметров \beta , которые минимизируют функцию потерь. Существует несколько методов для этого:

1. \*\*Градиентный спуск\*\*:

- Итеративный метод, который обновляет параметры, двигаясь в направлении, уменьшающем функцию потерь.

- На каждом шаге параметры обновляются по формуле:



где \( \alpha \) — скорость обучения (learning rate).

2. \*\*Стохастический градиентный спуск (SGD)\*\*:

- Вариант градиентного спуска, который обновляет параметры после обработки каждого отдельного наблюдения, а не всего набора данных. Это делает его более быстрым для больших данных.

3. \*\*Метод Ньютона (Newton's Method)\*\*:

- Использует вторые производные функции потерь для нахождения оптимальных параметров. Этот метод может быстрее сходиться, но требует больше вычислительных ресурсов.

4. \*\*Метод максимального правдоподобия\*\*:

- Ищет такие параметры, которые максимизируют вероятность наблюдения данных при данных параметрах модели.

### Пример

Представьте, что вы работаете в банке и хотите предсказать, получит ли клиент одобрение на кредит. У вас есть данные о клиентах, такие как их доход, возраст и кредитная история.

1. \*\*Формула\*\*:

- Сначала вы строите модель, используя логистическую функцию и данные о клиентах.

- Модель вычисляет вероятность \( P(\text{одобрение} | \text{данные клиента}) \).

2. \*\*Функция потерь\*\*:

- Вычисляете логистическую функцию потерь для оценки, насколько предсказанные вероятности соответствуют фактическим решениям об одобрении.

3. \*\*Обучение\*\*:

- Используете градиентный спуск, чтобы корректировать параметры модели, минимизируя функцию потерь.

После обучения модели вы сможете использовать её для предсказания вероятности одобрения кредита для новых клиентов, что поможет принимать более обоснованные решения.

1. Регуляризация. Назначение, виды, пример для линейной и логистической регрессии. В чем проявилась регуляризация в дз2

### Регуляризация

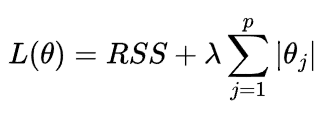
Регуляризация — это техника, используемая для предотвращения переобучения модели. Переобучение происходит, когда модель слишком хорошо подстраивается под обучающие данные, но плохо обобщает на новых, невиданных данных. Регуляризация добавляет штраф к функции потерь модели, что помогает контролировать сложность модели.

### Виды регуляризации

1. \*\*L1 Регуляризация (Лассо)\*\*

- \*\*Описание\*\*: L1 регуляризация добавляет к функции потерь сумму абсолютных значений весов модели, умноженную на гиперпараметр \( \lambda \).

- \*\*Формула\*\*:

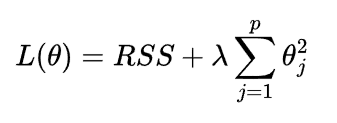


- \*\*Эффект\*\*:Способствует отбору признаков, так как может обнулять веса некоторых из них, оставляя только важные признаки.

2. \*\*L2 Регуляризация (Гребневая)\*\*

- \*\*Описание\*\*: L2 регуляризация добавляет к функции потерь сумму квадратов весов модели, умноженную на гиперпараметр \( \lambda \).

- \*\*Формула\*\*:



- \*\*Эффект\*\*: Уменьшает значения всех весов, что помогает избежать переобучения, но не обнуляет их полностью.

3. \*\*Elastic Net Регуляризация\*\*

- \*\*Описание\*\*: Комбинация L1 и L2 регуляризаций. Использует обе суммы — абсолютных значений и квадратов весов.

- \*\*Формула\*\*:

\[

L(\theta) = RSS + \lambda\_1 \sum\_{j=1}^{p} |\theta\_j| + \lambda\_2 \sum\_{j=1}^{p} \theta\_j^2

\]

- \*\*Эффект\*\*: Комбинирует преимущества обеих регуляризаций, может и обнулять веса, и уменьшать их значения.

### RSS (Residual Sum of Squares)

RSS означает "сумма квадратов остатков". Это мера, которая используется для оценки ошибки модели. Она вычисляется как сумма квадратов разниц между фактическими значениями и предсказанными моделью значениями.

Формула для RSS:

\[

RSS = \sum\_{i=1}^{n} (y\_i - \hat{y}\_i)^2

\]

Где:

- \( y\_i \) — фактическое значение зависимой переменной.

- \( \hat{y}\_i \) — предсказанное значение зависимой переменной.

### Примеры для линейной и логистической регрессии

#### Линейная регрессия

1. \*\*L1 регуляризация\*\*:

- \*\*Формула функции потерь\*\*:

\[

\text{Loss} = \text{MSE} + \lambda \|\mathbf{w}\|\_1

\]

- \*\*MSE\*\* (Mean Squared Error) — среднеквадратичная ошибка.

- \*\*Эффект\*\*: Отбор признаков, обнуляя веса некоторых из них.

2. \*\*L2 регуляризация\*\*:

- \*\*Формула функции потерь\*\*:

\[

\text{Loss} = \text{MSE} + \lambda \|\mathbf{w}\|\_2^2

\]

- \*\*Эффект\*\*: Уменьшение всех весов, предотвращение переобучения.

#### Логистическая регрессия

1. \*\*L1 регуляризация\*\*:

- \*\*Формула функции потерь\*\*:

\[

\text{Loss} = \text{Log\\_loss} + \lambda \|\mathbf{w}\|\_1

\]

- \*\*Log\\_loss\*\* — логистическая функция потерь.

- \*\*Эффект\*\*: Уменьшение значений всех весов, возможное обнуление некоторых из них.

2. \*\*L2 регуляризация\*\*:

- \*\*Формула функции потерь\*\*:

\[

\text{Loss} = \text{Log\\_loss} + \lambda \|\mathbf{w}\|\_2^2

\]

- \*\*Эффект\*\*: Уменьшение всех весов, предотвращение переобучения.

В ДЗ2 мы задавали коэффициент альфа, который учитывался при расчете вектора весов методом наименьших квадратов. Это коэффициент регуляризации, который влияет на степень штрафа, добавляемого к функции потерь, что помогает контролировать сложность модели и предотвращает переобучение.

7. Выбор модели. Отложенная выборка и кросс-валидация

8. Каким образом определялась наилучшая модель в полиномиальной регрессии в дз2. Привести схему разбиения датасета и построения графиков ошибок

9. Какие метрики использовались для оценки качества моделей в дз2, на каких этапах они вычислялись и для чего