Linear and Logistic Regression

!!! *Разглеждаме dataset-а като едно векторно пространство*

n(реда) **X** m(колони)

n - observation (една точка в m-мерното пространство)
m - feature (едно измерение в данните)

! По конвенция -> матрично умножение - ред по стълб

Linear Regression

1. Определение

Линейният регресионен анализ се използва за прогнозиране на стойността на дадена променлива въз основа на стойността на друга променлива.

Линейната регресия е основен статистически метод за анализ на зависимостта между променливи. Тя се използва за предсказване на стойността на зависима променлива (наречена **целева променлива** или **отговор**) въз основа на една или повече независими променливи (наречени **признаци** или **предиктори**). В основната си форма, линейната регресия предполага, че съществува линейна зависимост между целевата променлива и независимите променливи.

Изходната променлива (зависима променлива ,target) е число (става въпрос за регресия) и линейна комбинация на входните променливи с неизвестни коефициенти (в даден момент трябва да ни станат известни)

2. Пример:

Представи си, че искаш да предвидиш цената на жилище (y) в зависимост от неговата площ (x). Линейната регресия ще се опита да намери права линия, която най-добре описва тази зависимост, така че можеш да използваш модела за прогнозиране на цената на ново жилище въз основа на неговата площ.

3. Предимства на линейната регресия:

- **Простота**: Лесна за разбиране и интерпретация.
- **Ефективност**: Бързо се изпълнява дори при големи данни.
- **Интерпретируеми резултати**: Можеш лесно да разбереш как всяка независима променлива влияе на зависимата променлива.

4. Ограничения:

- **Линейност**: Предполага, че връзката между променливите е линейна, което не винаги е вярно в реалния живот.
- **Чувствителност към шум**: Моделът е чувствителен към аномалии и неточни данни.
- **Корелация на признаците**: Ако независимите променливи са силно корелирани помежду си (мултиколинеарност), това може да доведе до нестабилни коефициенти и трудности при интерпретацията.

5. Зависима и Независима променлива

В статистиката и анализа на данни, **зависима променлива** и **независима променлива** са термини, използвани за описание на връзката между различни променливи в даден модел.

5.1 Зависима променлива

Зависимата променлива (наричана още **отговор** или **целева променлива**) е тази, която се опитваме да предскажем или обясним. Тя зависи от стойностите на независимите променливи. С други думи, тя променя стойността си в зависимост от промените в независимите променливи.

5.2 Независима променлива

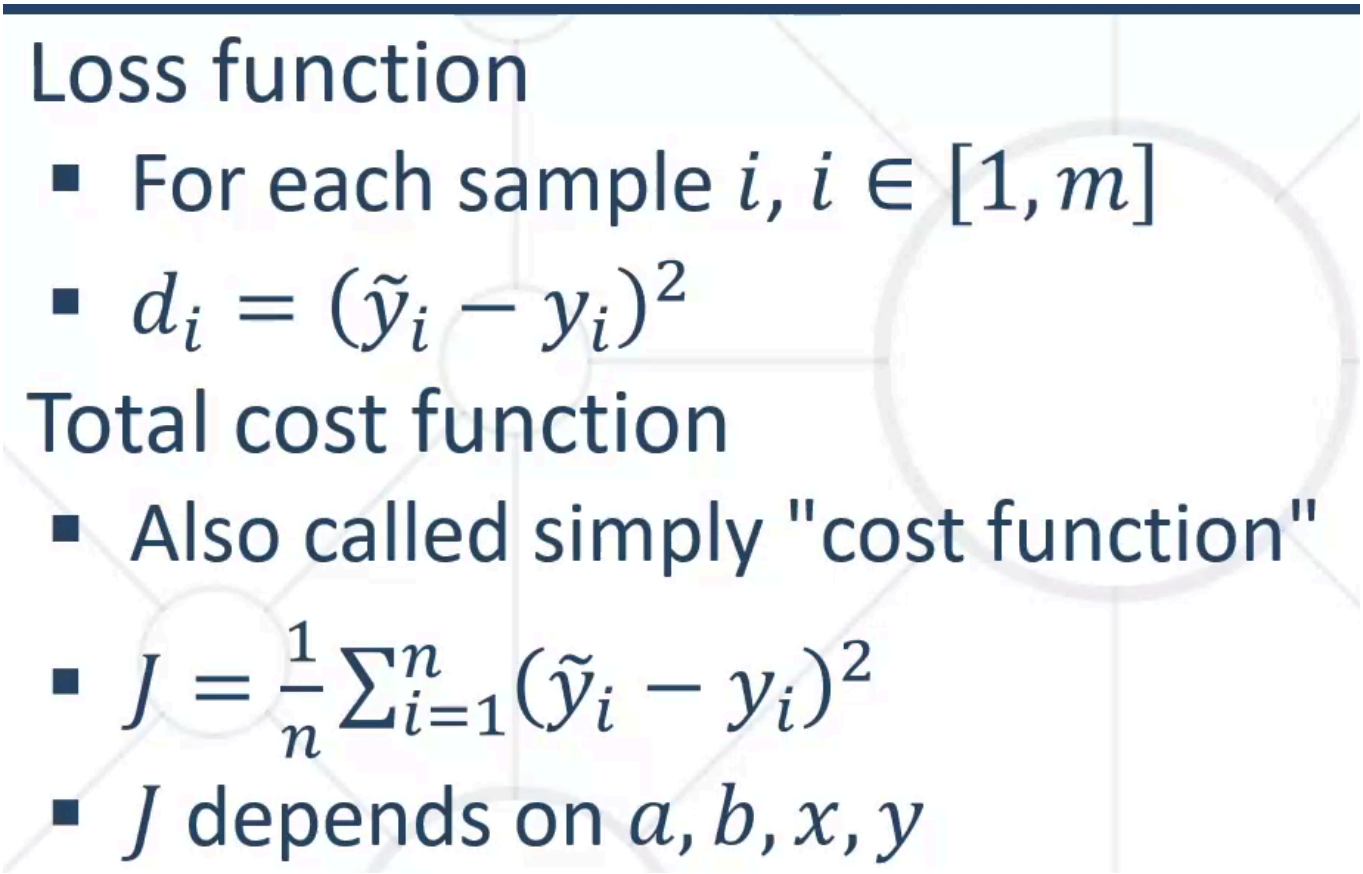
Независимата променлива (наричана още **фактор** или **предиктор**) е променливата, която оказва влияние върху зависимата променлива. Тя е независима в смисъл, че не зависи от другите променливи в модела, а влияе върху тях.

5.3 Пример:

Представи си, че правиш проучване за това как **времето за учене** влияе на **оценките на учениците** на изпит.

- **Зависима променлива**: Оценката на ученика на изпита (това е променливата, която искаме да предскажем).
- **Независима променлива**: Времето, което ученикът прекарва в учене (това е факторът, който оказва влияние върху оценката).

6. Метрики за разстояние



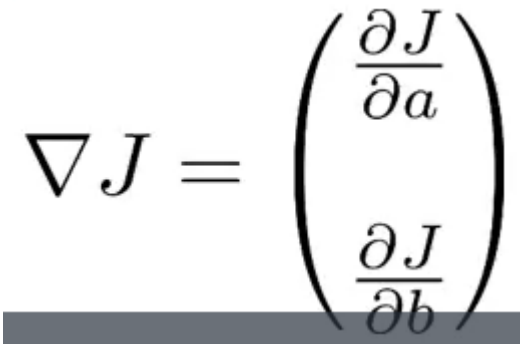
J - средна квадратична грешка(нашата метрика колко добре се справя моделът)

7. Gradiant descent

Един от най-популярните алгоритми за оптимизация, използван в машинното обучение и други области за минимизиране на функцията на загубата. Той помага да се намери минималната стойност на дадена функция, която представлява разликата между предсказаните и реалните стойности (грешката на модела).

Представи си, че се намираш на върха на хълм и искаш да слезеш до най-ниската точка (минимум). За да постигнеш това, можеш да следваш наклона на хълма (градиента) в посоката на най-стръмното слизане. Градиентният спуск прави точно това – той изчислява наклона на функцията (градиента) и се движи в посоката, противоположна на

него, за да намали стойността на функцията.



$$\nabla J = \begin{pmatrix} \frac{\partial J}{\partial a} \\ \\ \frac{\partial J}{\partial b} \end{pmatrix}$$

8. RANSAC

(Random Sample Consensus) - начин за изключване на outliers. Взимаме случаен брой данни и fit-ваме модел върху тях, след това взимаме друг subset и т.н. В някои от моделите ще има съгласие, в други не. Данните, които са били inliers във всеки един модел, дават нашия sample consensus.

9. Обяснения, функции, атрибути...

`MinMaxScaler` - (Preprocessing function, NOT A MODEL) взима всяка една колона и я трансформира в стойности между 0 и 1 (by default).

`score()` - изчислява R^2 (коефициент на детерминация)**, който е показател за качеството на модела. Този показател измерва каква част от вариацията в целевата променлива може да бъде обяснена от модела.
Стойности на R^2:

- R^2=1: Моделът перфектно обяснява всички вариации в данните.
- R^2**=0**: Моделът не обяснява никаква част от вариациите, и е толкова добър, колкото е средната стойност на целевата променлива.
- R^2<0: Моделът е по-лош от хоризонтална линия (просто средна стойност) и не е добър в обяснението на данните.
- **0<**R^2**<1**: Моделът обяснява част от вариацията, но не е перфектен.

`estimator_` - След като избере случайна подгрупа от данни, алгоритъмът RANSAC използва оценител, за да изчисли параметрите на модела. Това може да бъде различно в зависимост от проблема, който се решава. Например, ако се опитваме да напаснем права линия към данни, оценителят ще изчисли параметрите на тази линия (наклон и отместване).

`estimator_.coef_` - Атрибутът `coef_` съдържа коефициентите на регресионния модел, показващи колко се променя целевата променлива (y), когато независимата променлива се увеличи с 1 единица, при фиксирани останали променливи.

- Ако даден коефициент е **положителен**, това означава, че увеличението на съответната независима променлива води до **увеличение на y**.
- Ако коефициентът е **отрицателен**, това показва, че увеличаването на съответната променлива води до **намаление на y**.
- Пример:
  Коефициенти - [8.05526872, 0.58314896, -13.62191589] ->
  1. **8.05526872** – Първата променлива има положителен ефект върху y, като всяко увеличение с 1 единица води до увеличение на y с 8.06 единици.
  2. **0.58314896** – Втората променлива има по-слаб положителен ефект. Увеличение с 1 единица води до нарастване на y с 0.58 единици.
  3. **-13.62191589** – Третата променлива има силен **отрицателен ефект**. Увеличение с 1 единица намалява y с 13.62 единици.

`inlier_mask_` - показва кои от точките в набора от данни са **вътрешни (inliers)** спрямо намерената моделна линия или повърхнина. Това е булев масив (True/False) с размер, равен на броя на точките в обучаващия набор от данни. Всяка стойност в масива показва дали съответната точка от данните е била **inlier** спрямо модела:

- `True`: Точката е вътрешна (т.е. добре се вписва в модела).
- `False`: Точката е **външна (outlier)**, т.е. тя се отклонява значително от модела и не е използвана за финалната регресия.

Polynomial Regression

Полиномиалната регресия е форма на линейна регресия, при която връзката между независимите и зависимите променливи се моделира като полиномна функция от по-висока степен. Това е разширение на линейната регресия, която разглежда само линейни зависимости (прави линии). Полиномиалната регресия позволява моделирането на по-сложни зависимости, включително криволинейни връзки.

При полиномиалната регресия се добавят по-високи степени на независимата променлива x:

$$y = b0 + b1 \cdot x + b2 \cdot x^2 + b3 \cdot x^3 + \cdots + bn \cdot x^n$$

Logistic Regression

Логистичната регресия е статистически метод, който се използва за моделиране на бинарни изходи (т.е. когато изходът може да приеме само две стойности, като "да/не", "0/1", "истина/лъжа" и т.н.).

Основни характеристики:

1. **Изходът е дискретен (бинарен)**: Логистичната регресия се използва, когато искаме да предскажем дали нещо ще се случи или не (две възможности). Пример: ще купи ли потребителят даден продукт? (да/не).
2. **Функция на логистичната регресия (сигмоидна функция)**: Тъй като регресията работи с непрекъснати стойности, но ние имаме нужда от двоен резултат, логистичната регресия използва **сигмоидна функция** (наричана още логистична функция). Тази функция трансформира изхода от линейна регресия в стойност между 0 и 1, която след това може да се интерпретира като вероятност.
   Формата на логистичната функция е следната:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$