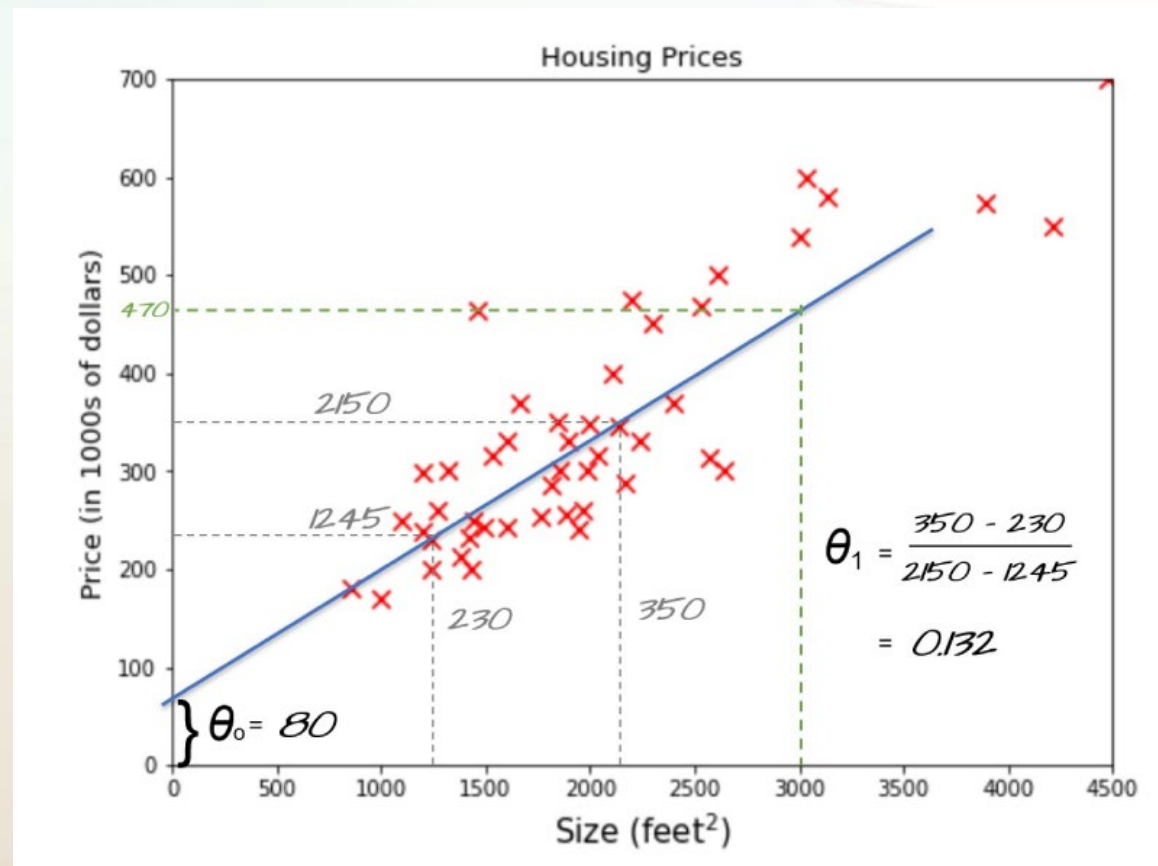


# Линейная регрессия

Основные понятия машинного обучения. Обучающая и валидационная выборка. Целевая переменная. Метрики, оценка качества. Функционал ошибки. Градиентный спуск.

# Задача линейной регрессии

- Наши **признаки**:
  - площадь квартиры
  - расстояние до метро
  - этаж
  - ...
- Наша **целевая переменная**:
  - цена квартиры
- Наша задача:
  - построить прямую так, чтобы для наших  $x$  ее  $y$  был максимально похож на правду



# Линейная регрессия: веса

- Цель: подобрать такие коэффициенты уравнения прямой, чтобы по нашим признакам можно было угадать примерный ответ (целевую переменную):

$$w_1x_1 + w_2x_2 + \dots + w_0 = y$$

- $x_1, x_2, \dots$  - это наши признаки (площадь квартиры, время до метро...)
- $y$  – это целевая переменная (цена квартиры)
- $w_1, w_2, \dots$  - это веса, или коэффициенты
- $w_0$  - это свободный коэффициент (шум)

# Как будем учить?

1. Возьмем случайные веса
2. Посчитаем предсказанные игреки для всех известных объектов
3. Сравним с правильными ответами
4. Поправим веса, чтобы наши игреки стремились к правильным ответам
5. Вернемся к пункту 2
6. ???
7. PROFIT!

# Как сравнивать игреки?

- Очевидно, нужно узнать, на сколько в среднем ошибается алгоритм, то есть:

$$\frac{1}{n} \sum_{i=1}^n (y_{pred} - y_{true})$$

- Что в такой формуле не нравится?

# Метрики оценки качества

- MSE:  $\frac{1}{n} \sum_{i=1}^n (y_{pred} - y_{true})^2$  и RMSE:  $\sqrt{\frac{1}{n} \sum_{i=1}^n (y_{pred} - y_{true})^2}$
- MAE:  $\frac{1}{n} \sum_{i=1}^n |y_{pred} - y_{true}|$
- Коэффициент детерминации:  $R^2(a, X) = 1 - \frac{\sum_{i=1}^l (a(x_i) - y_i)^2}{\sum_{i=1}^l (y_i - \hat{y})^2}$
- MSLE
- MAPE
- SMAPE
- ...

# Как подобрать веса?

- Возьмем, например, MSE: очевидно, что мы хотим, чтобы он был поменьше (чем меньше MSE, тем меньше ошибка модели)
- Следовательно, нам нужно **минимизировать функцию ошибки**
- То есть, уравнение, которое нам нужно решить (в матричной форме):

$$\frac{1}{n} ||X_w - y||^2 \rightarrow \min_w$$

- Это называется метод наименьших квадратов



# Аналитическое решение МНК

$$w = (X^T X)^{-1} X^T y$$

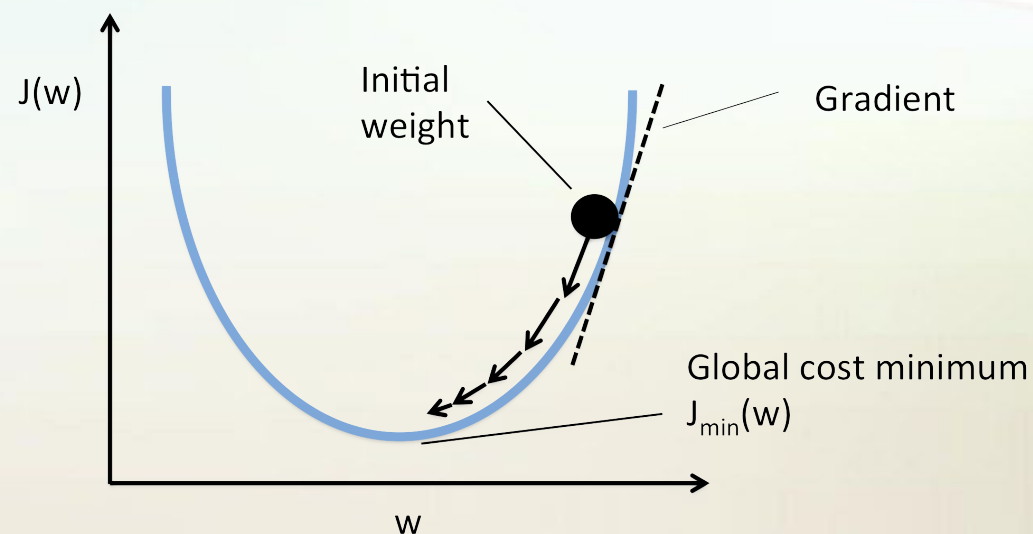
Недостатки:

- Обращение матрицы – сложная операция ( $O(n^3)$  от числа признаков)
- Матрица  $X^T X$  может быть вырожденной или плохо обусловленной
- Если функционал ошибки будет другим, можем вообще не решить задачу



# Градиентный спуск

- Градиент – вектор, в направлении которого функция растет
- Антиградиент – вектор, противоположный градиенту
- Если будем двигаться в направлении антиградиента, найдем минимум
- *(Вспоминайте Лагутина)*



# Градиентный спуск

- Пусть у нас только один вес  $w$  (для простоты)
- Инициализируем вес случайным числом:  $w^{(0)}$
- При добавлении к весу антиградиента  $-\frac{\partial Q}{\partial w}$  функция  $Q(w)$  убывает.
- Вычисляем конкретные значения производной для каждого объекта в выборке
- Считаем среднее арифметическое из них
- Вычитаем из веса  $w^{(0)}$
- Повторяем с начала

# Градиентный спуск

- Если у нас несколько весов, то делаем это для каждого из них.
- Общая формула изменения веса:

$$w^{(k)} = w^{(k-1)} - \nabla Q(w^{(k-1)})$$

- Обычно еще добавляют коэффициент, чтобы сразу весь градиент не вычитался:

$$w^{(k)} = w^{(k-1)} - \eta \nabla Q(w^{(k-1)})$$

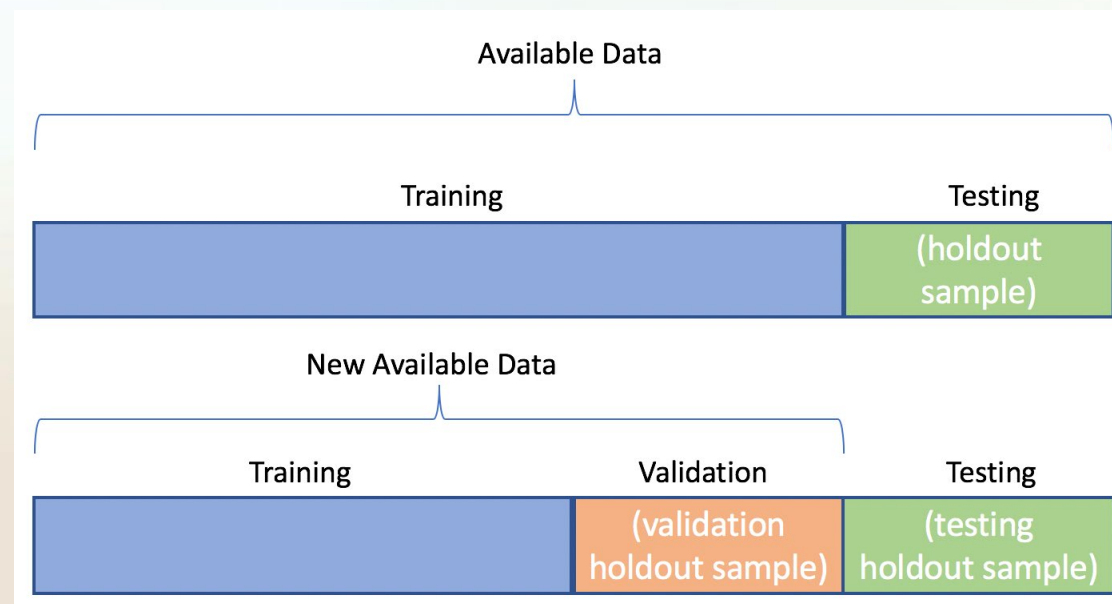
- Этот параметр называется learning rate. Мы еще много будем про него говорить на курсе по нейронкам

# Как проверить качество?

- Допустим, мы обучили наш алгоритм. Как удостовериться, что он хорошо работает?
- Очевидно, считаем все те же метрики
- Метрика качества может быть такая же, как функция ошибки, а может быть другой
- Но нельзя ее считать на той выборке, на которой мы учились: это будет нечестно

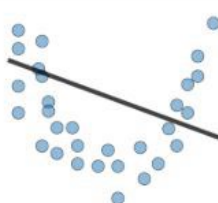


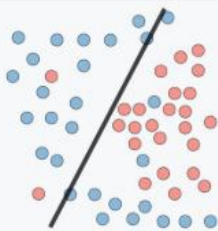
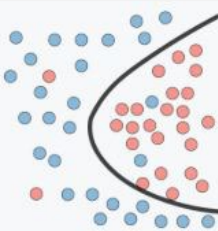
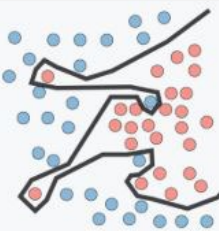

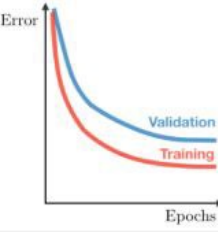
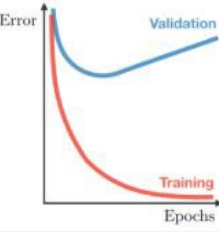
# Как проверить качество?

- Следовательно, перед обучением нужно отложить какое-то количество данных, чтобы модель их не видела



# Переобучение и недообучение

- В алгоритмах классического МО очень важна работа с признаками
- Если признаки линейно зависимы между собой, то высок риск переобучения
- Работа с фичами – это искусство

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"><li>• High training error</li><li>• Training error close to test error</li><li>• High bias</li></ul>	<ul style="list-style-type: none"><li>• Training error slightly lower than test error</li></ul>	<ul style="list-style-type: none"><li>• Very low training error</li><li>• Training error much lower than test error</li><li>• High variance</li></ul>
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	<ul style="list-style-type: none"><li>• Complexify model</li><li>• Add more features</li><li>• Train longer</li></ul>		<ul style="list-style-type: none"><li>• Perform regularization</li><li>• Get more data</li></ul>



# Наконец - практическая часть!

устанавливаем scikit learn, если еще не!

```
pip install scikit-learn
```

```
(conda install -c anaconda scikit-learn)
```