Конспект по теме "Линейная регрессия изнутри"

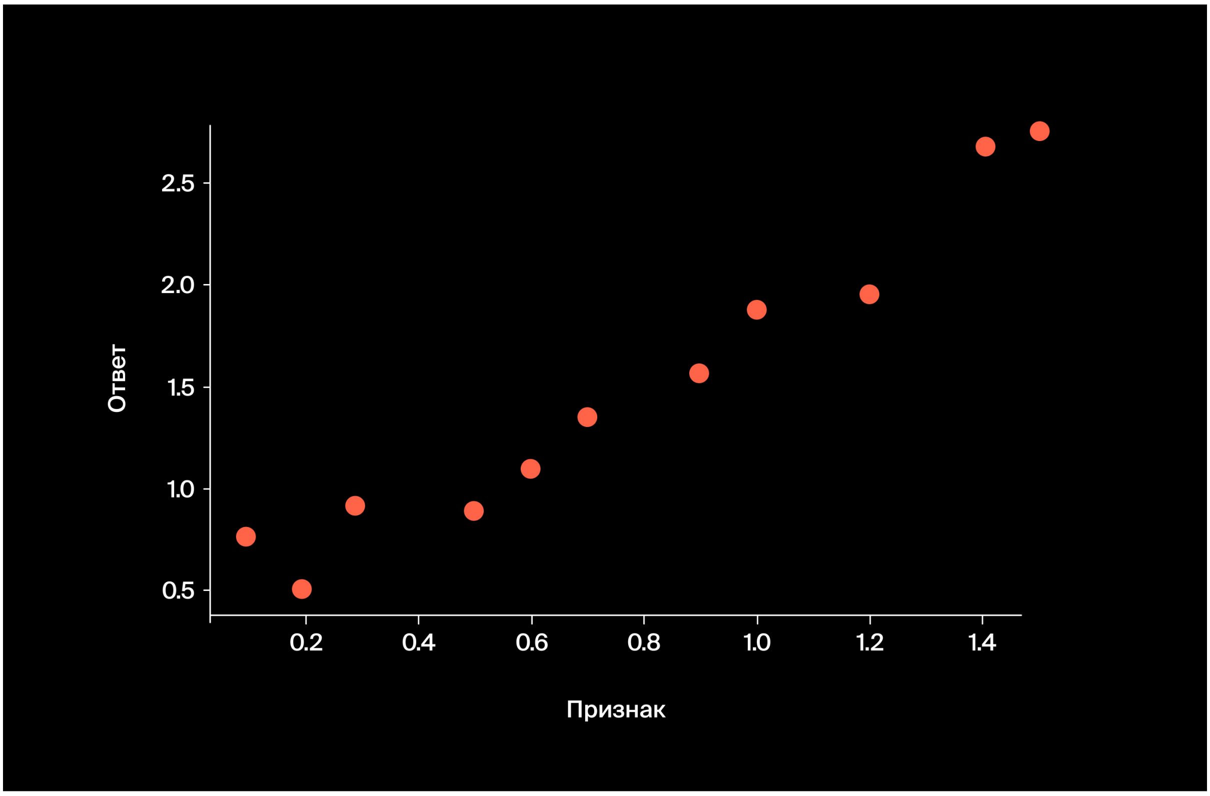
Модель линейной регрессии

У линейной регрессии признаки — это вектор чисел в *n*-мерном пространстве (допустим, *x*). Предсказание модели (*a)* вычисляется так: скалярно умножается вектор признаков на вектор **весов** (*w*), затем к этому произведению прибавляется величина **сдвига** предсказания (*bias*):

$$a = (x, w) + w_0$$

Вектор *w* и скаляр *w0* — это параметры модели. В векторе *w* параметров *n*, а в *w0* — один. То есть количество параметров больше длины вектора признаков на единицу.

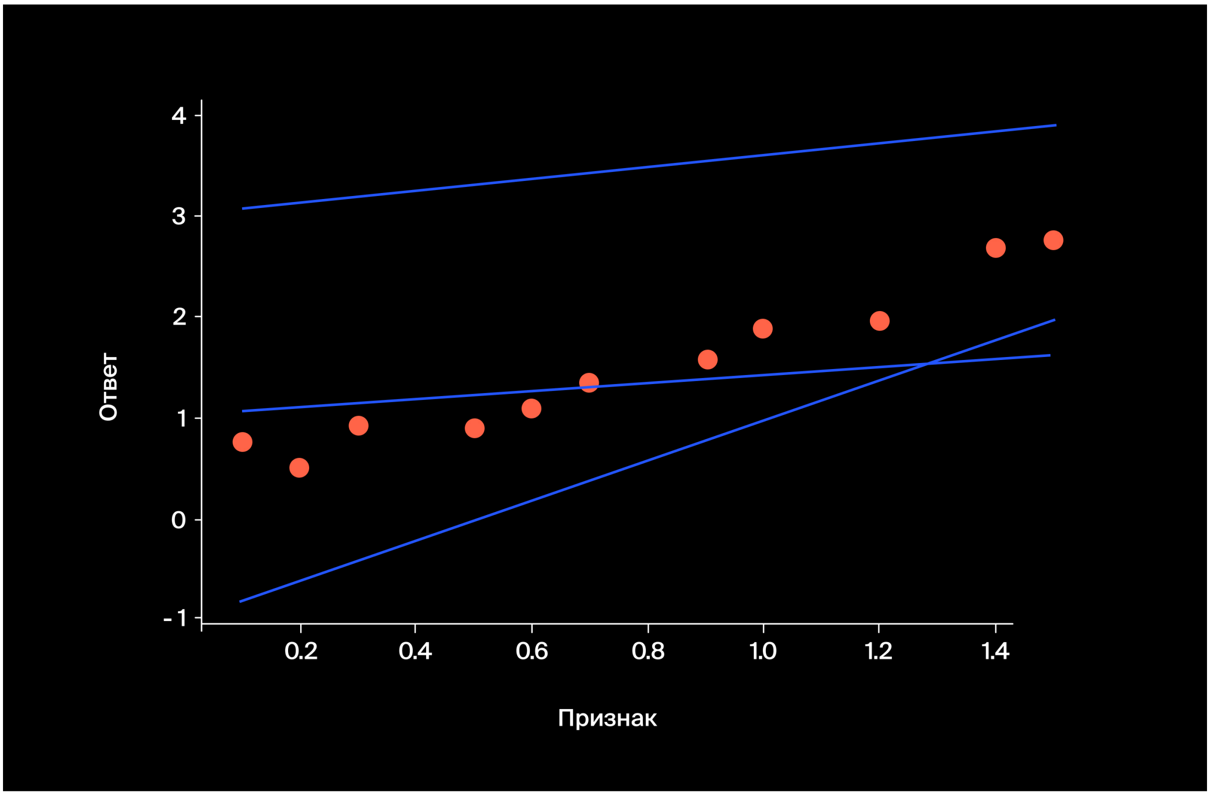
Если длина вектора признаков равна единице, то в выборке всего один признак. Изобразим этот признак с ответами на графике:



Графики предсказаний для линейной регрессии задаются уравнением:

$$y = wx + w_0$$

Если менять параметры *w* и *w0*, получается любая прямая, или линия (отсюда и название модели):



Задача обучения

Разберём алгоритм обучения. Нашей метрикой качества будет *MSE:* её наименьшее значение модель должна получить на тестовых данных. Задачу обучения сформулируем такую: найти параметры модели, при которых значение **функции потерь** (*loss function*) на обучающей выборке минимально. Как метрики качества, она принимает на вход правильные ответы и предсказания. А возвращает число, которое называют «потерями» (их-то и нужно минимизировать). В нашей задаче эта функция приравнивается к *MSE*. Но обычно функция потерь применяется для обучения, а метрика качества — для тестирования.

Задачу обучения запишем в векторном виде. Обучающую выборку представим как матрицу *X*, в которой строки соответствуют объектам, а столбцы — признакам. Параметры линейной регрессии обозначим *w* и *w0*. Чтобы получить вектор предсказаний *a*, умножим матрицу *X* на вектор *w* и прибавим величину сдвига *w0*.

Формула выглядит так:

$$a = Xw + w_0$$

Для сокращения записи изменим обозначения. В матрицу *X* добавим столбец, состоящий только из единиц (он идёт нулевым); а параметр *w0* — к вектору w:

$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

$$(w_1, w_2, \dots, w_n) \longrightarrow (w_0, w_1, w_2, \dots, w_n)$$

Затем умножим матрицу *X* на вектор *w*. Сдвиг умножится на вектор из единиц (нулевой столбец). Получим такой вектор предсказаний *a:*

$$a = Xw$$

Введём новое обозначение *y* — вектор значений целевого признака для обучающей выборки.

Запишем формулой задачу обучения линейной регрессии для функции потерь *MSE:*

$$w = \arg\min_{w} MSE(Xw, y)$$

Функция *argmin()* находит минимум и возвращает, при каком аргументе он был достигнут.

Обратная матрица

Единичная матрица (*E*) — это квадратная матрица, на главной диагонали которой стоят единицы, а остальные элементы — нули. Если любую матрицу *A* умножить на единичную (или наоборот), получится эта же матрица *A*:

$$AE = EA = A$$

Обратная для квадратной матрицы *A* — матрица *A* с верхним индексом -1, произведение которой на *A* равно единичной матрице. Умножение может быть в любом порядке:

$$AA^{-1} = A^{-1}A = E$$

Матрицы, для которых можно найти обратные, называются **обратимыми**. Но не у каждой матрицы есть обратная. Такие матрицы называются **необратимыми**. Необратимые матрицы встречаются редко. Если сгенерировать случайную матрицу функцией *numpy.random.normal()*, вероятность получить необратимую матрицу близка к нулю.

Чтобы найти обратную матрицу, вызовите функцию *numpy.linalg.inv()*. Также она поможет проверить матрицу на обратимость: если матрица необратима, будет обнаружена ошибка.

Обучение линейной регрессии

Задача обучения линейной регрессии такая:

$$w = \arg\min_{w} MSE(Xw, y)$$

Минимальное значение *MSE* получается, когда веса равны этой величине:

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

Как получили эту формулу:

- Транспонированная матрица признаков умножается на себя;
- Вычисляется обратная к результату матрица;
- Обратная умножается на транспонированную матрицу признаков;
- Результат умножается на вектор значений целевого признака.