Microsoft

DevCon School

Технологии будущего

DevCon School

Технологии будущего

Линейная и логистическая регрессия от Excel через Python к Azure ML

Комаров Михаил

Microsoft MVP

Цели

Линейная регрессия

Вы узнаете то-то

Вы разберетесь в этом-то

Логистическая регрессия

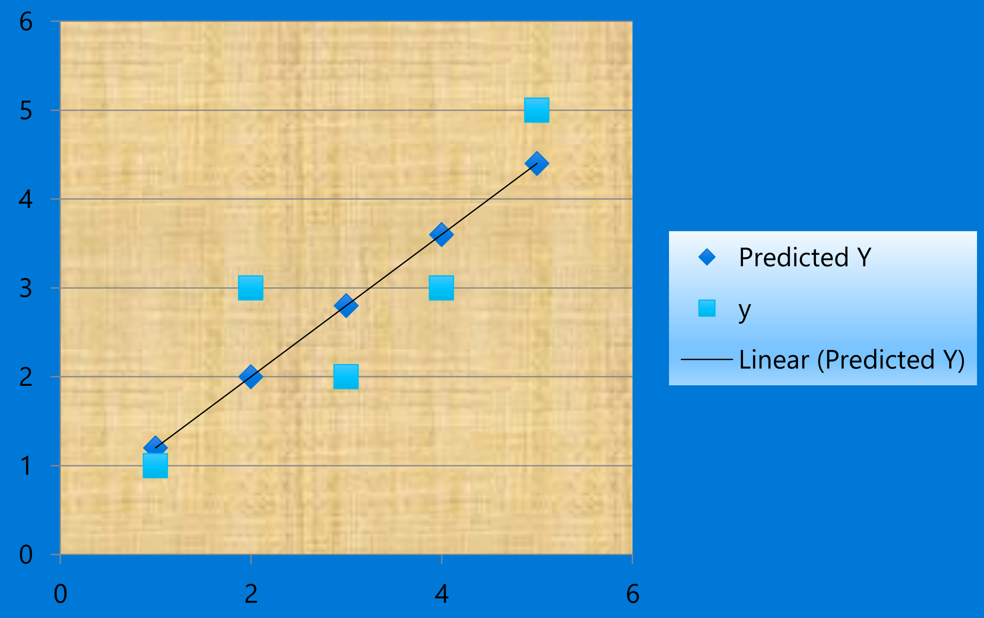
Вы научитесь тому-то

Метрики качества

Вас будет беспокоить вот это

$$RSS = \sum_i (y_i - (a + bx_i))^2$$

Линейная регрессия



#msdevcon

DevCon School
Технологии будущего

Линейная регрессия: коэффициенты

Результат расчёта

$$\begin{cases} a = \bar{y} - b\bar{x} \\ b = \dfrac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - (\bar{x})^2} \end{cases}$$

Минимизируемая функция

$$RSS = \sum_i (y_i - (a + bx_i))^2$$

Поиск стационарных точек для RSS

$$\begin{cases} \dfrac{\partial RSS}{\partial a} = \sum_i 2(y_i - a - bx_i) = 0 \\ \dfrac{\partial RSS}{\partial b} = \sum_i 2(y_i - a - bx_i)x_i = 0 \end{cases}$$

$$\begin{cases} \sum_i y_i - na - b\sum_i x_i = 0 \\ \sum_i x_i y_i - a\sum_i x_i - b\sum_i x_i^2 = 0 \end{cases}$$

$$\begin{cases} \bar{y} - a - b\bar{x} = 0 \\ \overline{xy} - a\bar{x} - b\overline{x^2} = 0 \end{cases} \quad \begin{cases} a = \bar{y} - b\bar{x} \\ \overline{xy} - (\bar{y} - b\bar{x})\bar{x} - b\overline{x^2} = 0 \end{cases} \quad \begin{cases} a = \bar{y} - b\bar{x} \\ \overline{xy} - \bar{x}\bar{y} + b\left[(\bar{x})^2 - \overline{x^2}\right] = 0 \end{cases}$$

Линейная регрессия: коэффициенты r и R2

Коэффициент детерминации R^2

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{RSS}{TSS} = \frac{ESS}{TSS}$$

$$RSS + ESS = TSS$$

$$RSS = \sum_i (y_i - \hat{y}_i)^2$$

residual sum of squares (сумма квадратов отклонений)

$$TSS = \sum_i (y_i - \bar{y})^2$$

total sum of squares (общая сумма квадратов)

$$ESS = \sum_i (\hat{y}_i - \bar{y})^2$$

explained sum of squares (объяснённая сумма квадратов)

Коэффициент корреляции Пирсона $r_{y,\hat{y}}$

$$r_{y,\hat{y}} = \frac{\sum_i (y_i - \bar{y})(\hat{y} - \bar{y})}{\sqrt{\sum_i (y_i - \bar{y})^2 \sum_i (\hat{y}_i - \bar{y})^2}} = \frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{y})}{\sqrt{TSS \cdot ESS}}$$

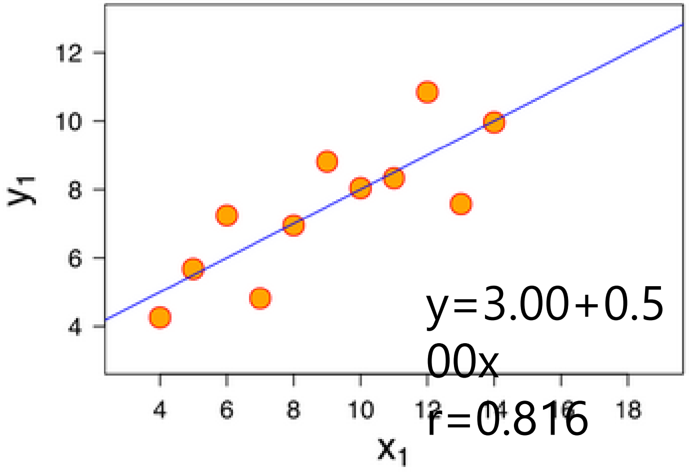
$$\hat{y}_i = a + bx_i$$

Связь между R^2 и $r_{y,\hat{y}}$

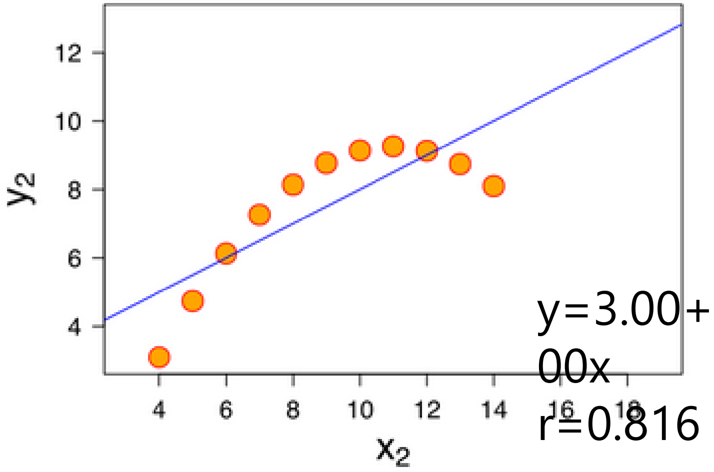
0 (т.к. МНК)

$$r_{y,\hat{y}} = \frac{\sum_i (y_i - \hat{y}_i + \hat{y}_i - \bar{y})(\hat{y}_i - \bar{y})}{\sqrt{TSS \cdot ESS}} = \frac{\sum_i (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) + \sum_i (\hat{y}_i - \bar{y})^2}{\sqrt{TSS \cdot ESS}} = \sqrt{\frac{ESS}{TSS}} = \sqrt{R^2}$$

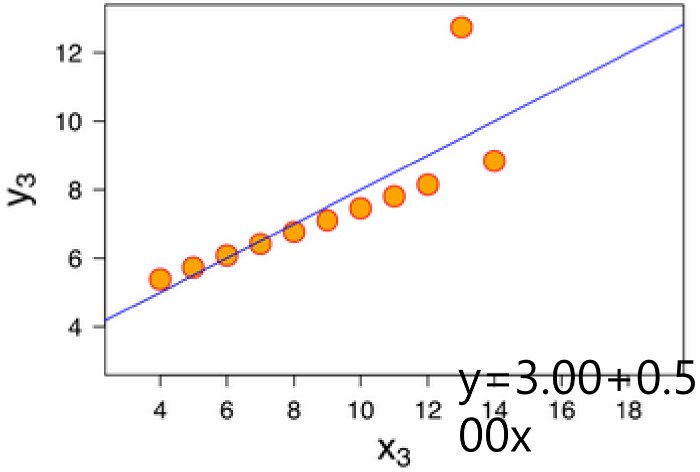
Квартет Энскомба (Anscombe’s quartet)



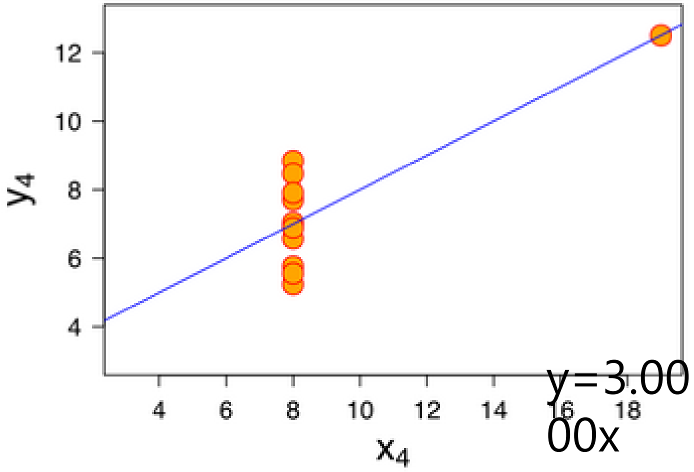
y=3.00+0.5
00x
r=0.816



y=3.00+0.5
00x
r=0.816

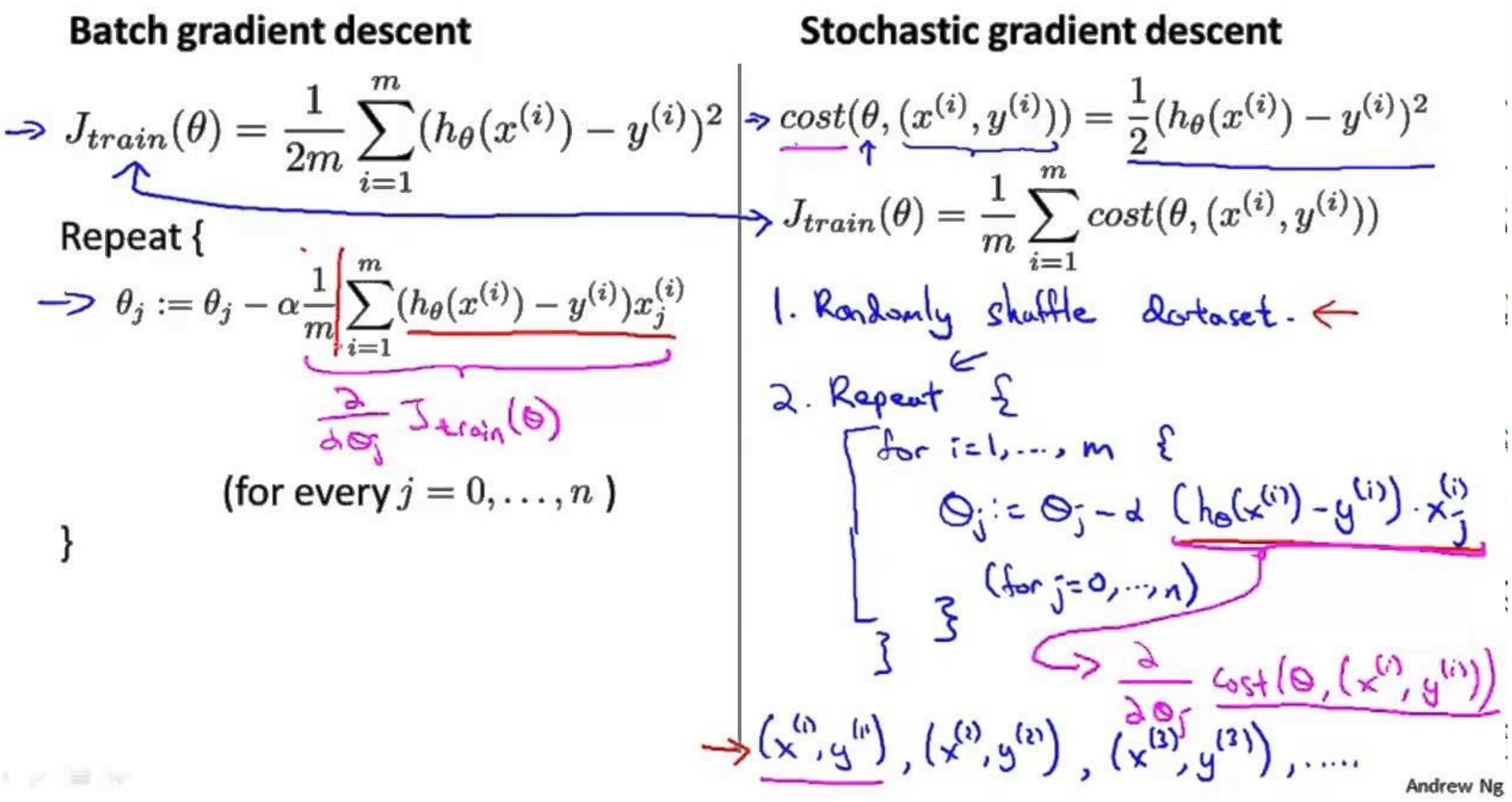


y=3.00+0.5
00x
r=0.816



y=3.00+0.5
00x
r=0.816

Градиентный спуск



Batch gradient descent

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Repeat {

$$\theta_j := \theta_j - \alpha\frac{1}{m}\underbrace{\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}}_{\frac{\partial}{\partial\theta_j}J_{train}(\theta)}$$

(for every $j = 0, \ldots, n$)

}

Stochastic gradient descent

$$cost(\theta, (x^{(i)}, y^{(i)})) = \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J_{train}(\theta) = \frac{1}{m}\sum_{i=1}^{m} cost(\theta, (x^{(i)}, y^{(i)}))$$

1. Randomly shuffle dataset.
2. Repeat {

for $i = 1, \ldots, m$ {

$$\theta_j := \theta_j - \alpha\underbrace{(h_\theta(x^{(i)}) - y^{(i)})\cdot x_j^{(i)}}_{\frac{\partial}{\partial\theta_j}cost(\theta,(x^{(i)},y^{(i)}))}$$

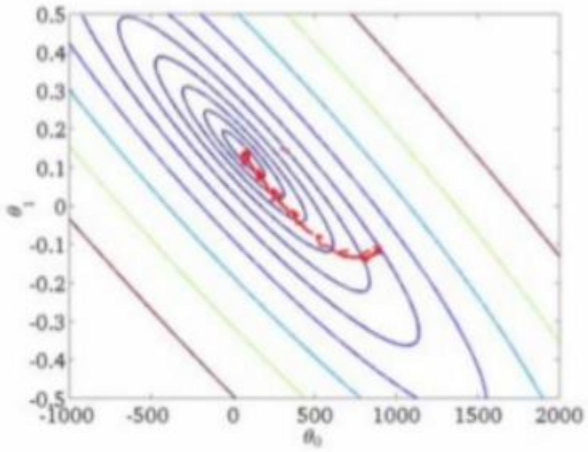
(for $j = 0, \ldots, n$)

}

}

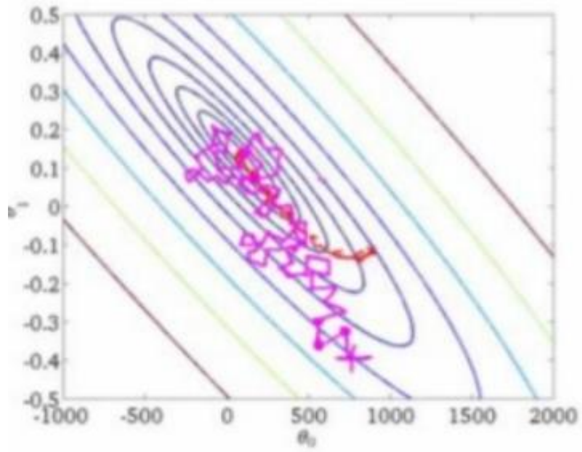
$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots$

Andrew Ng



Batch: gradient

$$x \leftarrow x - \eta\nabla F(x)$$



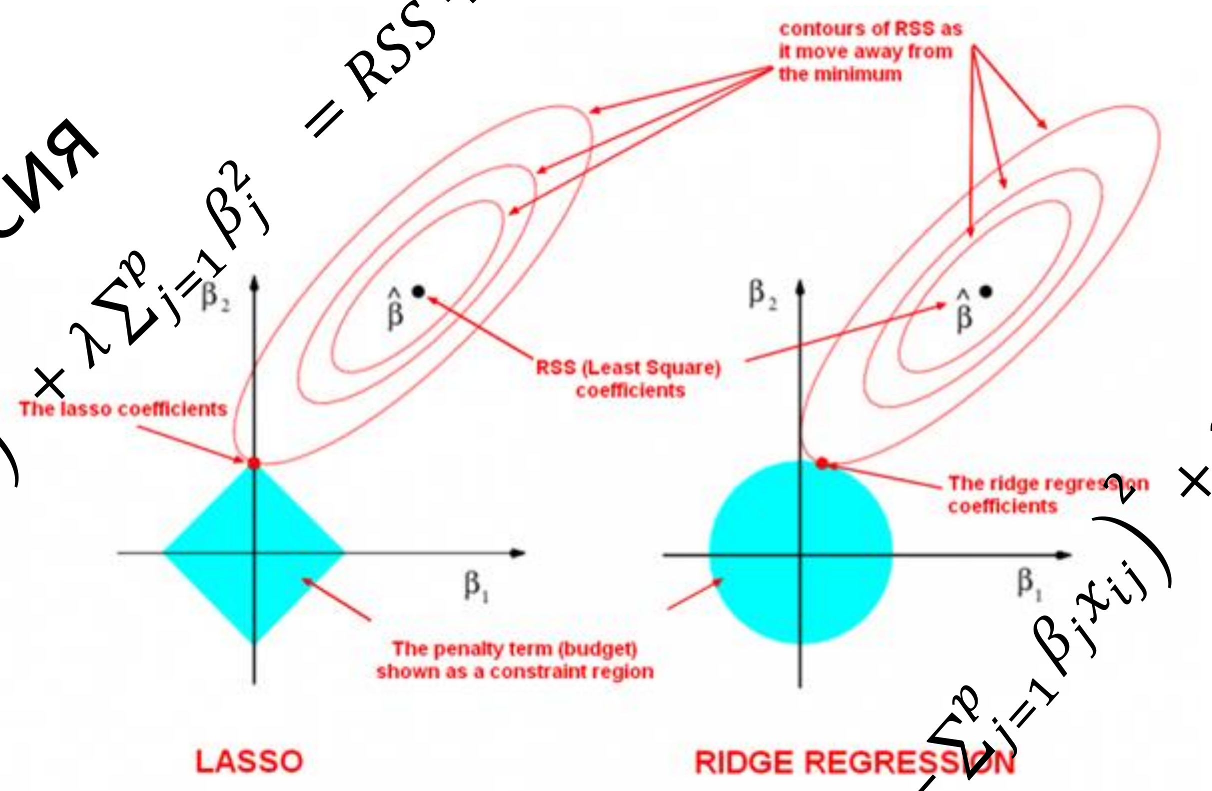
Stochastic: single-example gradient

$$x \leftarrow x - \eta\nabla F_i(x)$$

Регуляризация

Lasso регрессия

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2 = RSS+\lambda\sum_{j=1}^{p}\beta_j^2$$



$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = RSS+\lambda\sum_{j=1}^{p}|\beta_j|$$

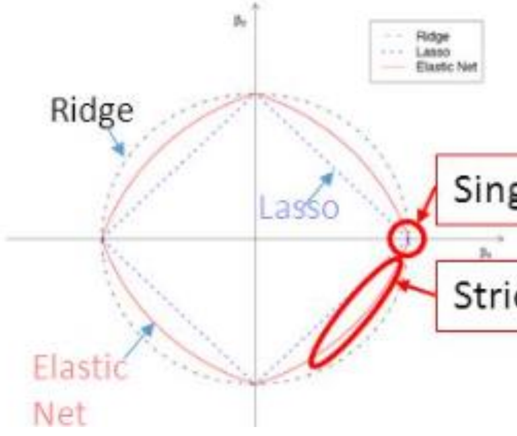
Ridge регрессия

Elastic Net

- *Elastic Net* penalize the size of the regression coefficients based on both their l^1 norm and their l^2 norm :

$$argmin_{\beta}\sum_{i}(y_i-\boldsymbol{\beta}'\boldsymbol{x}_i)^2+\lambda_1\sum_{k=1}^{K}|\beta_k|+\lambda_2\sum_{k=1}^{K}\beta_k^2$$

- The l^1 norm penalty generates a sparse model.
- The l^2 norm penalty:
  - Removes the limitation on the number of selected variables.
  - Encourages grouping effect.
  - Stabilizes the l^1 regularization path.



Geometric Illustration of Elastic Net, Ridge regression, and LASSO

Singularities at the vertexes (necessary for sparsity)

Strict convex edges (necessary for grouping)

Демонстрация

Линейная регрессия

Excel, Python, Azure ML

#msdevcon

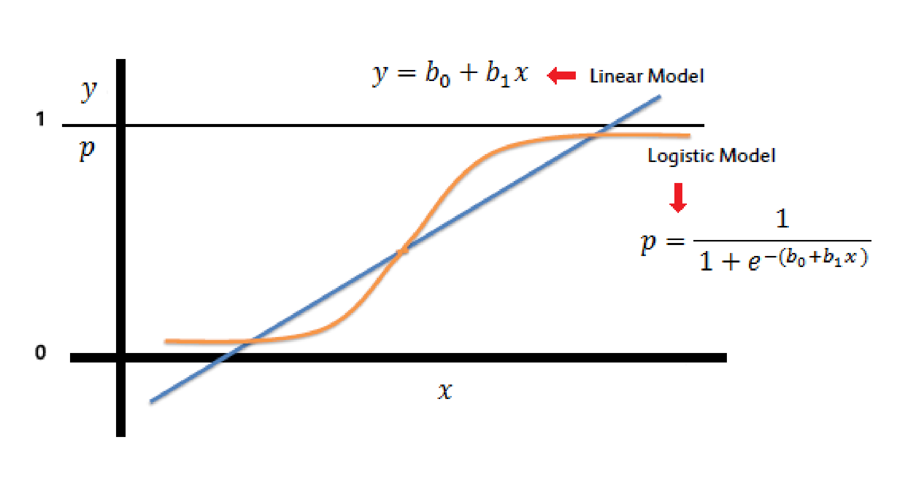
DevCon School
Технологии будущего

Практика

Работаем с линейной регрессией

Материалы:

https://www.sendspace.com/file/7razqd



Логистическая регрессия

Логистическая регрессия

Логистическая регрессия применяется для предсказания вероятности возникновения некоторого события по значениям множества признаков. Для этого вводится так называемая *зависимая переменная* y, принимающая лишь одно из двух значений — как правило, это числа 0 (событие не произошло) и 1 (событие произошло), и множество *независимых переменных* (также называемых признаками, предикторами или регрессорами) — вещественных $x_1, x_2, \ldots, x_n$, на основе значений которых требуется вычислить вероятность принятия того или иного значения зависимой переменной.

Делается предположение о том, что вероятность наступления события $y = 1$ равна:

$$\mathbb{P}\{y = 1 \mid x\} = f(z),$$

где $z = \theta^T x = \theta_1 x_1 + \ldots + \theta_n x_n$, x и θ — векторы-столбцы значений независимых переменных $x_1, \ldots, x_n$ и параметров (коэффициентов регрессии) — вещественных чисел $\theta_1, \ldots, \theta_n$, соответственно, а $f(z)$ — так называемая *логистическая функция* (иногда также называемая сигмоидом или логит-функцией):

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

Так как y принимает лишь значения 0 и 1, то вероятность первого возможного значения равна:

$$\mathbb{P}\{y = 0 \mid x\} = 1 - f(z) = 1 - f(\theta^T x).$$

Для краткости функцию распределения y при заданном x можно записать в таком виде:

$$\mathbb{P}\{y \mid x\} = f(\theta^T x)^y (1 - f(\theta^T x))^{1-y}, \quad y \in \{0, 1\}.$$

Фактически, это есть распределение Бернулли с параметром, равным $f(\theta^T x)$.

Подбор параметров

Для подбора параметров $\theta_1, \dots, \theta_n$ необходимо составить обучающую выборку, состоящую из наборов значений независимых переменных и соответствующих им значений зависимой переменной y. Формально, это множество пар $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$, где $x^{(i)} \in \mathbb{R}^n$ — вектор значений независимых переменных, а $y^{(i)} \in \{0, 1\}$ — соответствующее им значение y. Каждая такая пара называется обучающим примером.

Обычно используется метод максимального правдоподобия, согласно которому выбираются параметры θ, максимизирующие значение функции правдоподобия на обучающей выборке:

$$\hat{\theta} = \operatorname{argmax}_\theta L(\theta) = \operatorname{argmax}_\theta \prod_{i=1}^{m} \mathbb{P}\{y = y^{(i)} \mid x = x^{(i)}\}.$$

Максимизация функции правдоподобия эквивалентна максимизации её логарифма:

$$\ln L(\theta) = \sum_{i=1}^{m} \log \mathbb{P}\{y = y^{(i)} \mid x = x^{(i)}\} = \sum_{i=1}^{m} y^{(i)} \ln f(\theta^T x^{(i)}) + (1 - y^{(i)}) \ln(1 - f(\theta^T x^{(i)})).$$

Для максимизации этой функции может быть применён, например, метод градиентного спуска. Он заключается в выполнении следующих итераций, начиная с некоторого начального значения параметров θ:

$$\theta := \theta + \alpha \nabla \ln L(\theta) = \theta + \alpha \sum_{i=1}^{m} (y^{(i)} - f(\theta^T x^{(i)})) x^{(i)}, \alpha > 0.$$

На практике также применяют метод Ньютона и стохастический градиентный спуск.

Регуляризация

Для улучшения обобщающей способности получающейся модели, то есть уменьшения эффекта переобучения, на практике часто рассматривается логистическая регрессия с регуляризацией.

Регуляризация заключается в том, что вектор параметров θ рассматривается как случайный вектор с некоторой заданной априорной плотностью распределения $p(\theta)$. Для обучения модели вместо метода наибольшего правдоподобия при этом используется метод максимизации апостериорной оценки, то есть ищутся параметры θ, максимизирующие величину:

$$\prod_{i=1}^{m} \mathbb{P}\{y^{(i)} \mid x^{(i)}, \theta\} \cdot p(\theta).$$

В качестве априорного распределения часто выступает многомерное нормальное распределение $\mathcal{N}(0, \sigma^2 I)$ с нулевым средним и матрицей ковариации $\sigma^2 I$, соответствующее априорному убеждению о том, что все коэффициенты регрессии должны быть небольшими числами, идеально — многие малозначимые коэффициенты должны быть нулями. Подставив плотность этого априорного распределения в формулу выше, и прологарифмировав, получим следующую оптимизационную задачу:

$$\sum_{i=1}^{m} \log \mathbb{P}\{y^{(i)} \mid x^{(i)}, \theta\} - \lambda \|\theta\|^2 \ \to \max,$$

где $\lambda = \mathrm{const}/\sigma^2$ — параметр регуляризации. Этот метод известен как L2-регуляризованная логистическая регрессия, так как в целевую функцию входит L2-норма вектора параметров для регуляризации.

Если вместо L2-нормы использовать L1-норму, что эквивалентно использованию распределения Лапласа, как априорного, вместо нормального, то получится другой распространённый вариант метода — L1-регуляризованная логистическая регрессия:

$$\sum_{i=1}^{m} \log \mathbb{P}\{y^{(i)} \mid x^{(i)}, \theta\} - \lambda \|\theta\|_1 \ \to \max.$$

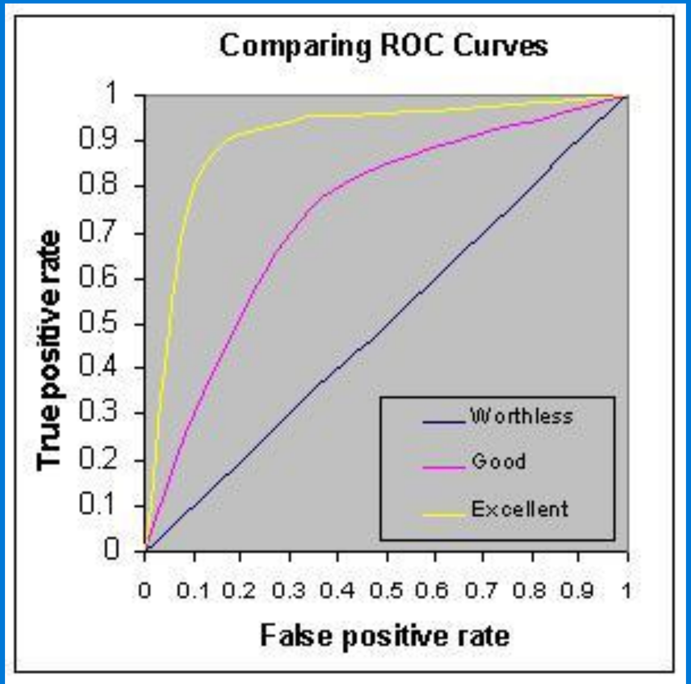
Практика

Работаем с логистической регрессией

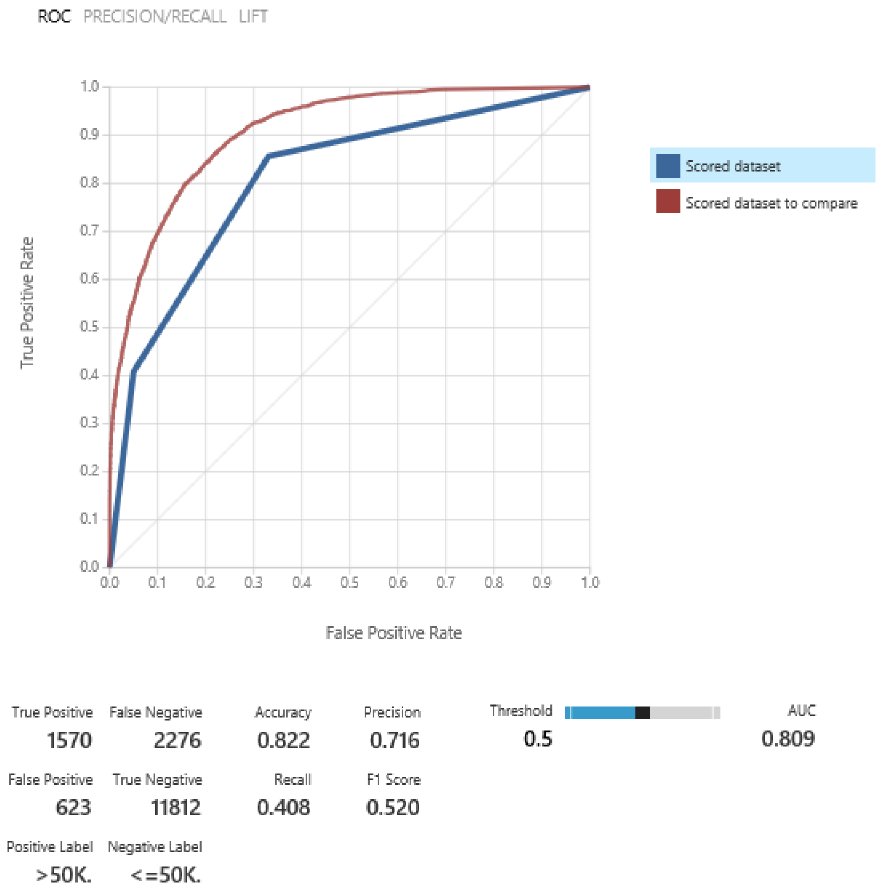
#msdevcon

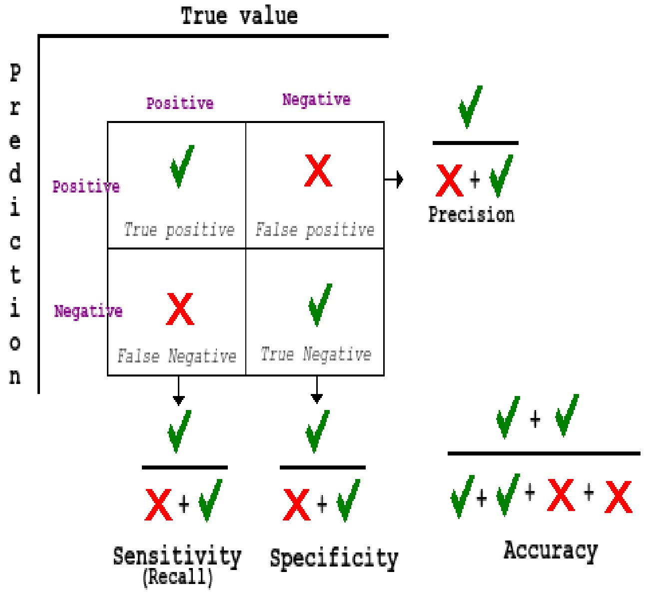
DevCon School
Технологии будущего

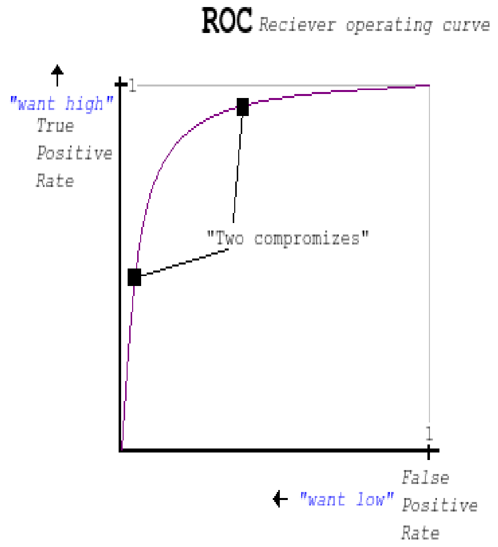
Метрики качества



Метрики качества

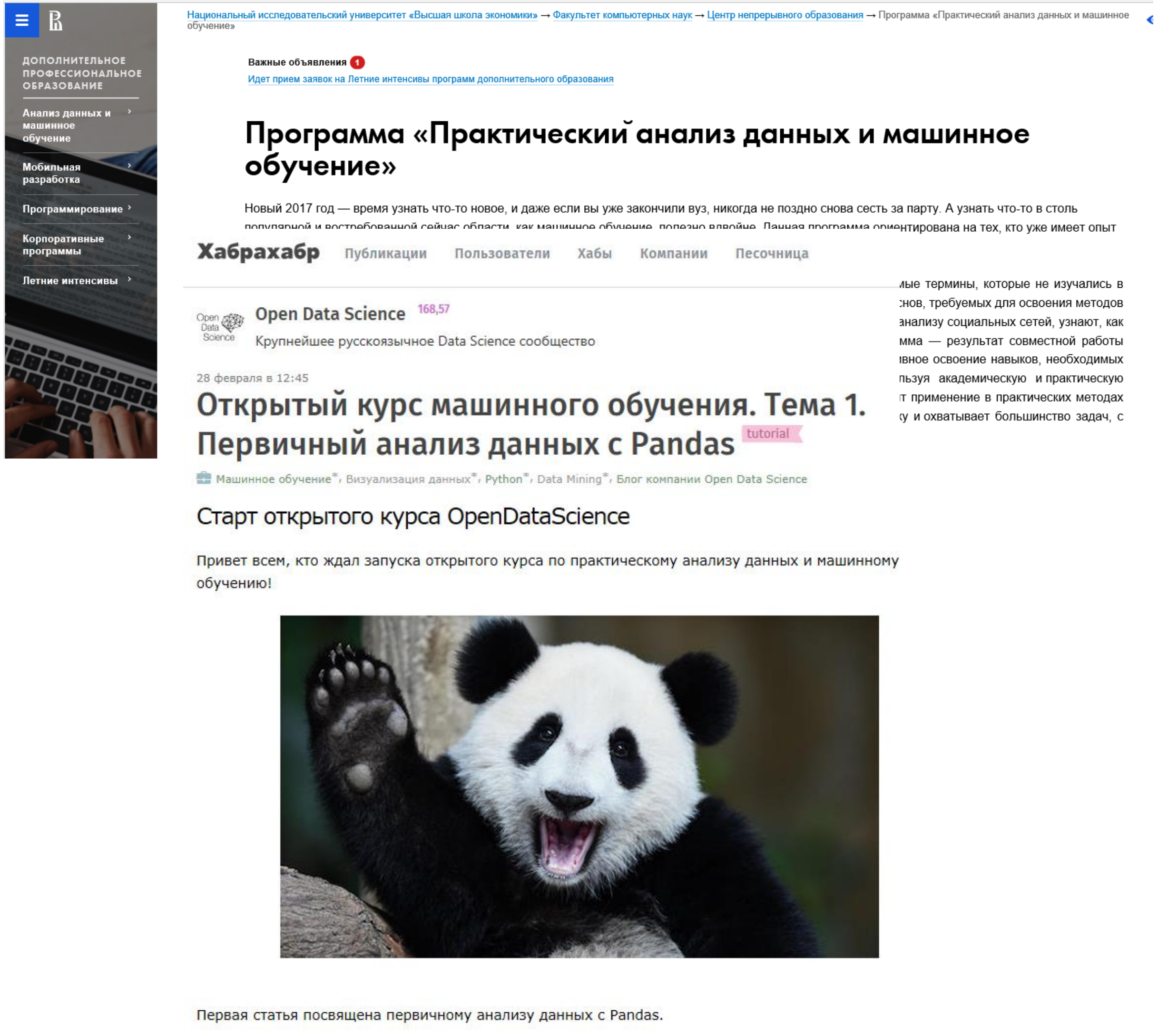


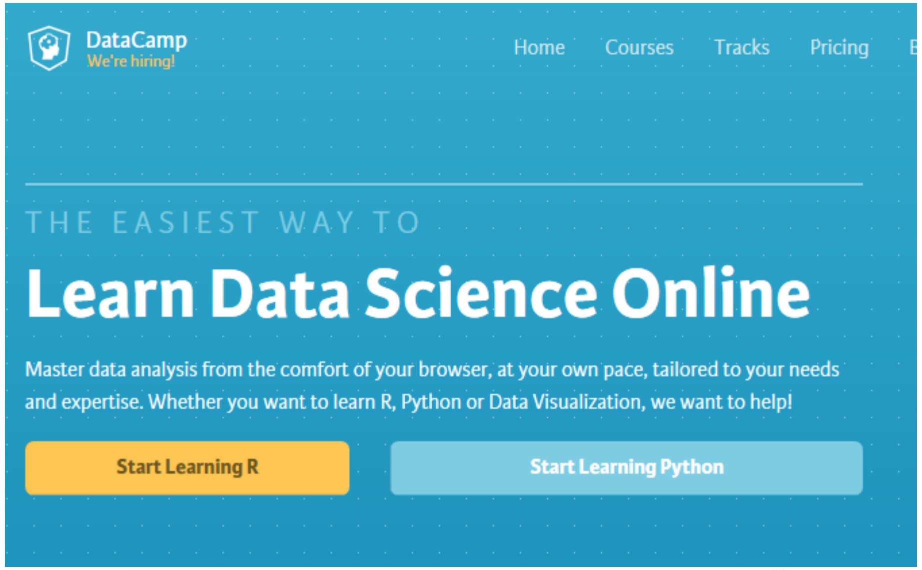


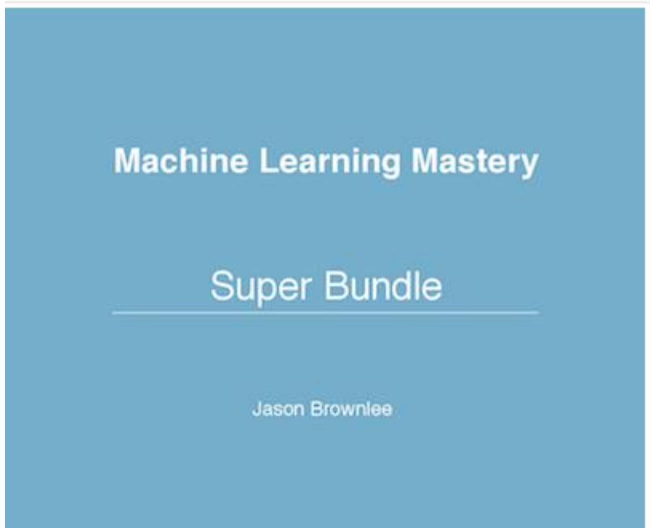


$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Ресурсы на которые я потратил ресурсы ☺







Полезные ресурсы

Читаем по Azure (книга)

https://blogs.msdn.microsoft.com/microsoft_press/2016/09/01/free-ebook-microsoft-azure-essentials-fundamentals-of-azure-second-edition/

Читаем по Azure ML (книга)

https://aka.ms/AzureML_pdf

Смотрим курсы по треку DS от Microsoft (edx)

https://academy.microsoft.com/en-us/professional-program/data-science/

Блог по AzureML от Microsoft

https://blogs.technet.microsoft.com/machinelearning/

Machine Learning Part 1 | SciPy 2016 Tutorial | Andreas Mueller & Sebastian Raschka

https://www.youtube.com/watch?v=OB1reY6IX-o&list=PLYx7XA2nY5Gf37zYZMw6OqGFRPjB1jCy6&index=1&t=988s

Q&A

Линейная и логистическая регрессия от Excel через Python к Azure ML

Комаров Михаил
Microsoft MVP

#msdevcon

DevCon School
Технологии будущего

Помогите нам стать лучше!

На вашу почту отправлена индивидуальная ссылка на электронную анкету. 3 июня в 23:30 незаполненная анкета превратится в тыкву.

Заполните анкету и подходите к стойке регистрации за приятным сюрпризом!

#msdevcon

Оставляйте отзывы в социальных сетях. Мы все читаем. Спасибо вам! ☺

Microsoft

Спасибо за внимание !!!