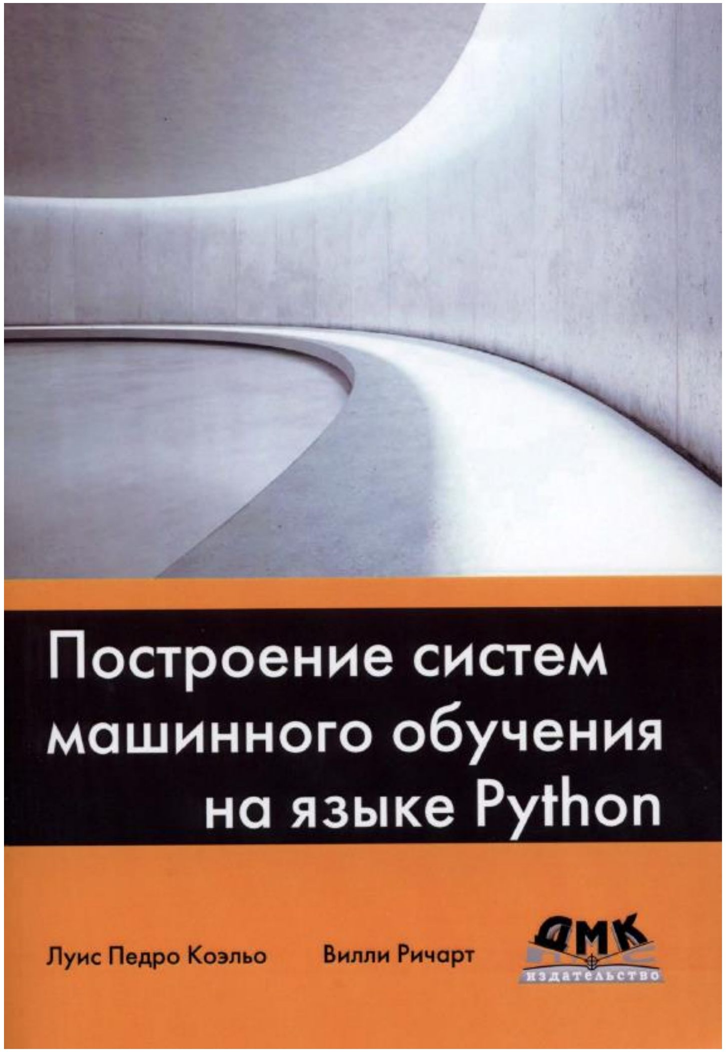
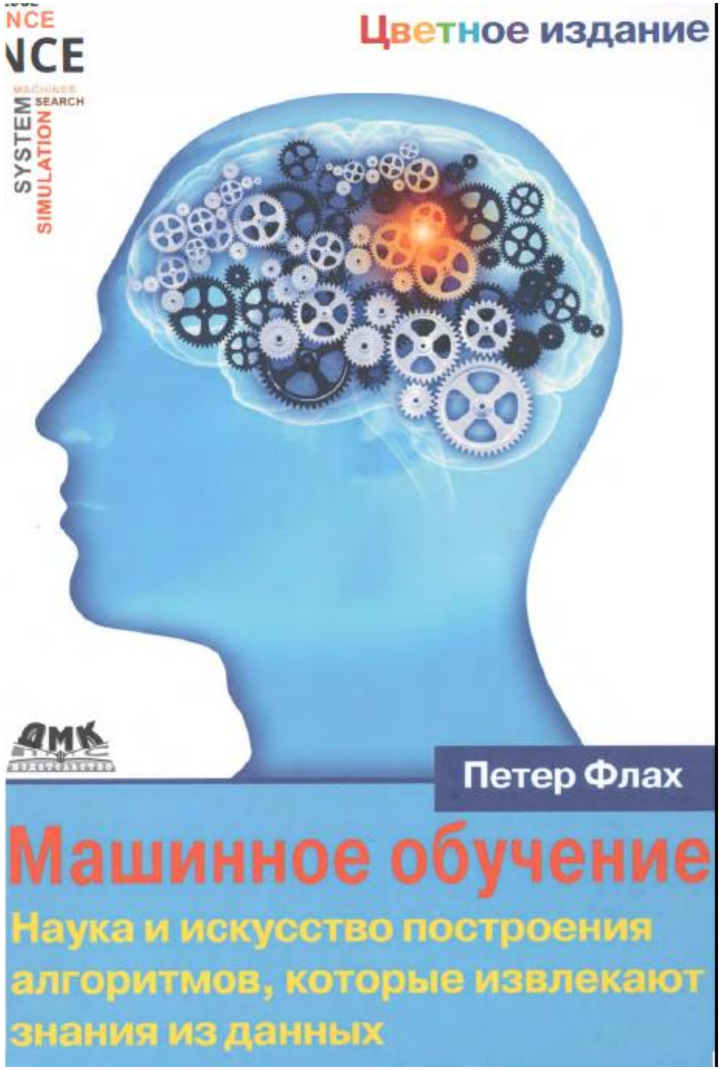
ЛИНЕЙНАЯ КЛАССИФИКАЦИЯ

Логистическая регрессия

ЛИТЕРАТУРА


Построение систем машинного обучения на языке Python
Луис Педро Коэльо
Вилли Ричарт
ДМК издательство


Цветное издание
NCE
NCE
SEARCH
SYSTEM
SIMULATION
ДМК
Петер Флах
Машинное обучение
Наука и искусство построения алгоритмов, которые извлекают знания из данных

Бинарный линейный классификатор

Дана обучающая выборка

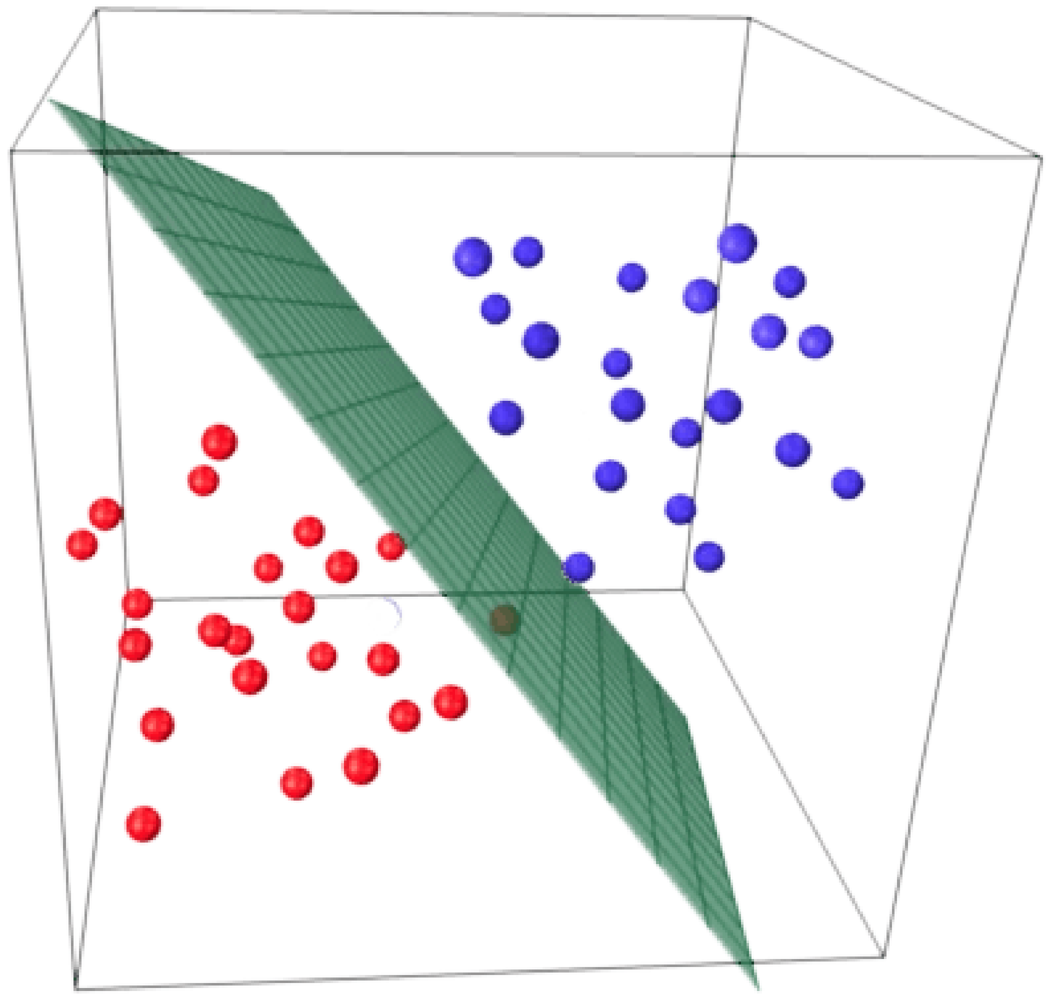
$X_N=\{(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\}, \quad \boldsymbol{x}_i \in \boldsymbol{R}^P,\ y_i \in \{-1,+1\}$

Цель: каждый новый входной вектор x отнести к одному их двух классов – положительному «+1» или отрицательному «+1»

$$\hat{y} = \hat{y}(x,w) = sign\left(w_0 + \sum_{j=1}^{p} w_j\, x_j\right) = sign(w^{\top}\boldsymbol{x}),$$

$$\boldsymbol{x} = (1, x_1, \ldots, x_p)$$

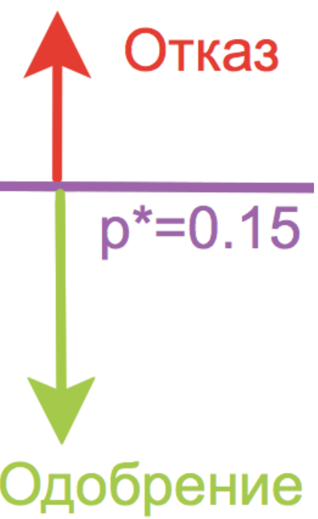
Линейная модель классификации



Логистическая регрессия как линейный классификатор

- Логистическая регрессия прогнозирует вероятность $\boldsymbol{p}_+$ отнесения примера $\boldsymbol{x}$ к классу "+1“.
- Пример: банковский скоринг

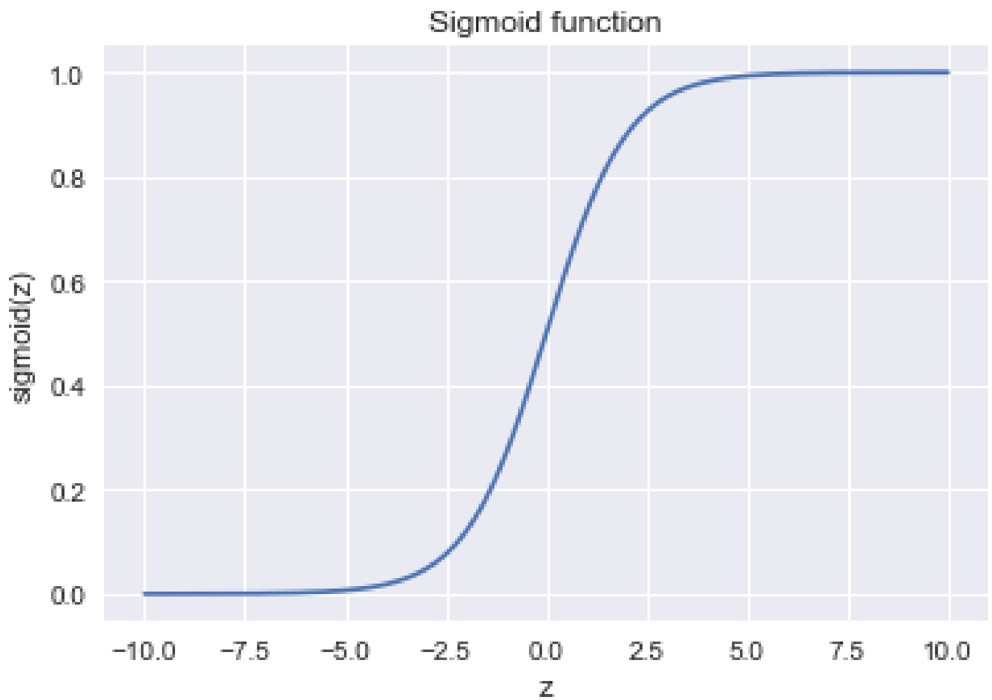
Клиент	**Вероятность невозврата**
Mike	0.78
Jack	0.45
Larry	0.13
Kate	0.06
William	0.03
Jessica	0.02



Логистическая регрессия

$$p_+ = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

где $z = w^{\top}\boldsymbol{x} = w_0 + \sum_{j=1}^{p} w_j\, x_j$

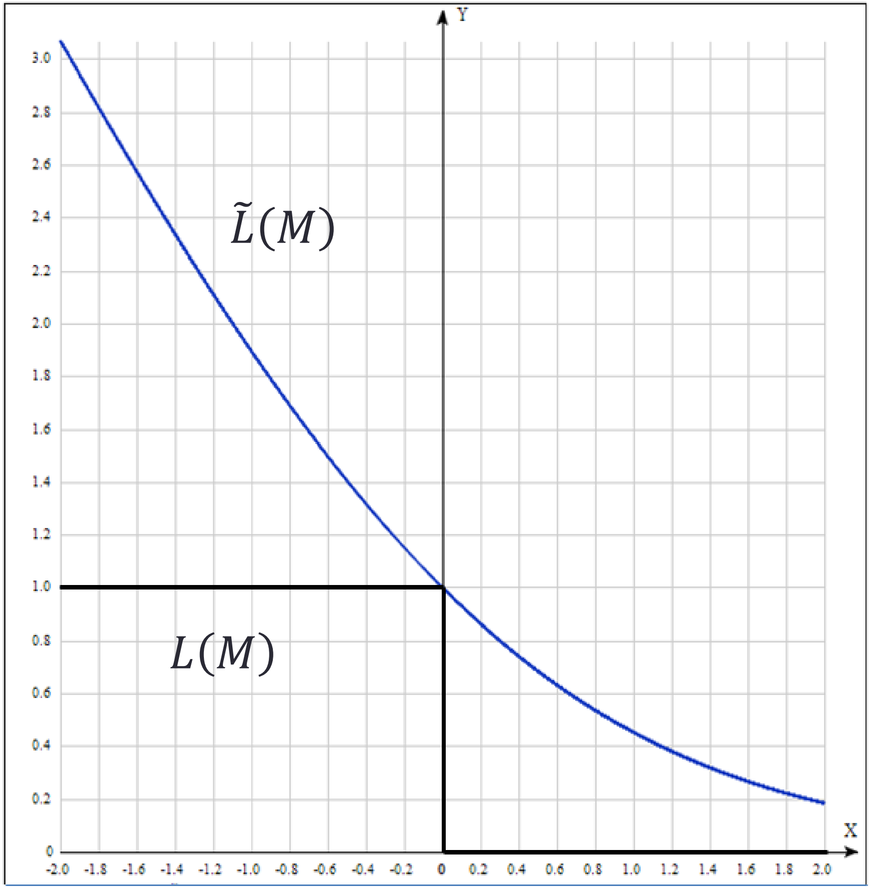


Функция потерь (ошибок классификации)

- Доля неправильных ответов:
- $E(W) = \frac{1}{N}\sum_{i=1}^{N}[y_i \neq \hat{y}_i] \; = \frac{1}{N}\sum_{i=1}^{N}[sign(w^\intercal \boldsymbol{x}_i) \neq y_i]$
- $E(W) = \frac{1}{N}\sum_{i=1}^{N}[y_i(w^\intercal \boldsymbol{x}_i) < 0]$
- $M_i = y_i(w^\intercal \boldsymbol{x}_i)$ - отступ
- $L(M) = [M < 0] - \text{пороговая функция}$

Верхняя оценка

$$L(M) \leq \tilde{L}(M) = log_2(1 + \exp(-M))$$



Логистическая функция потерь

- $ERR(w) = \sum_{i=1}^{N} log_2\left(1 + \exp\left(-y_i(w^\intercal \boldsymbol{x}_i)\right)\right)$

С учетом L2-регуляризации:

- $ERR(w) = \sum_{i=1}^{N} log_2\left(1 + \exp\left(-y_i(w^\intercal \boldsymbol{x}_i)\right)\right) +$

$+ \frac{1}{\lambda} \sum_{j=1}^{p} {w_j}^2$

Использование полиномиальных признаков для нелинейного разделения

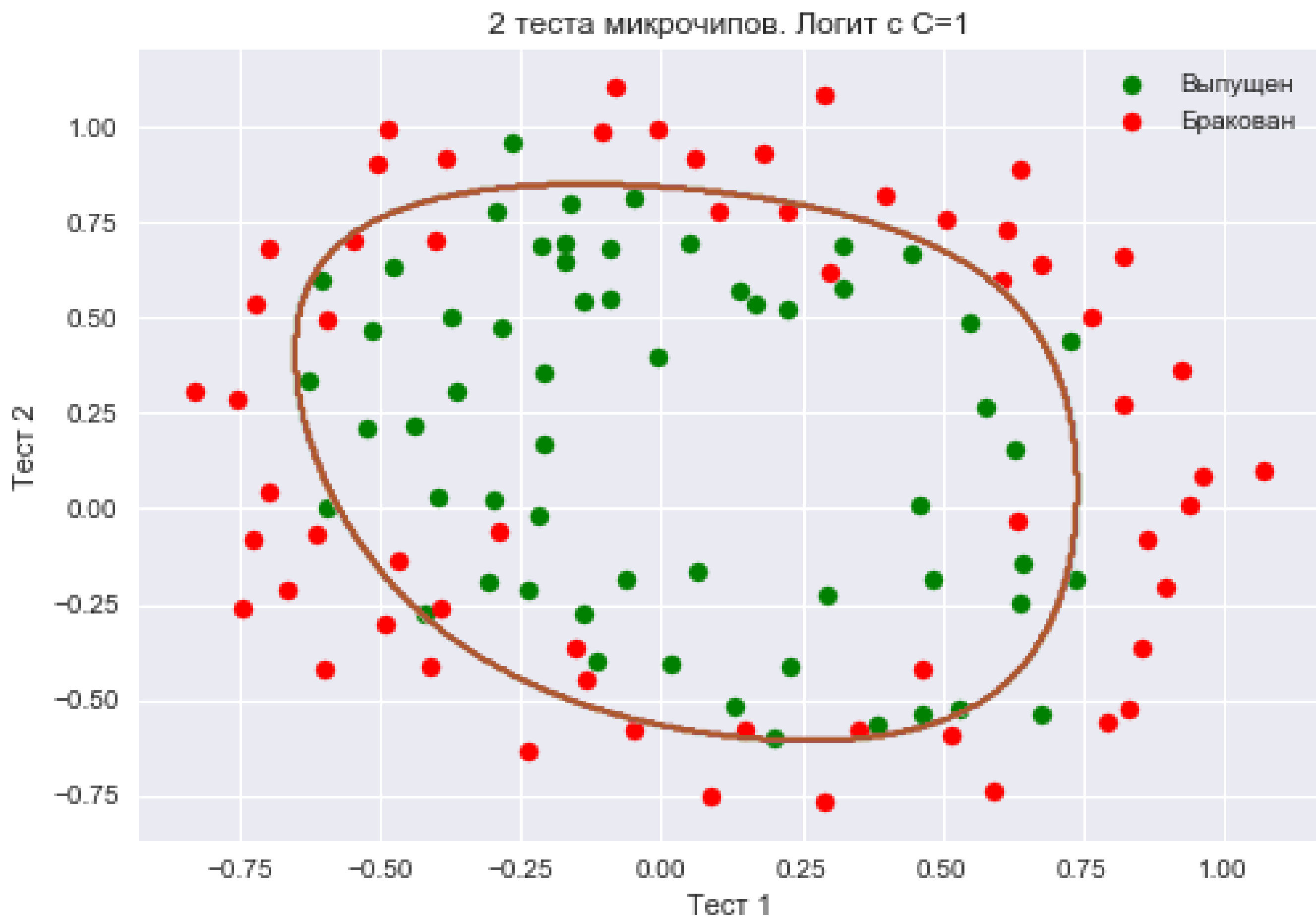
Полиномиальными признаками до степени d для двух переменных x_1 и x_2 мы называем следующие:

$$\{x_1^d, x_1^{d-1}x_2, \ldots x_2^d\} = \{x_1^i x_2^j\}_{i+j=d, i,j\in\mathbb{N}}$$

Например, для $d = 3$ это будут следующие признаки:

$$1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3$$

Пример нелинейного разделения классов



Confusion matrix (матрица ошибок классификации)

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Здесь $\hat{y}$ — это ответ алгоритма на объекте, а y — истинная метка класса на этом объекте.

Таким образом, ошибки классификации бывают двух видов: False Negative (FN) и False Positive (FP).

Метрики качества классификации

- Доля правильных ответов: $accuracy = \dfrac{TP+TN}{TP+TN+FP+FN}$

Малоинформативна в задачах с неравными классами.

Пример. Допустим, мы хотим оценить работу спам-фильтра почты. У нас есть 100 не-спам писем, 90 из которых наш классификатор определил верно, и 10 спам-писем, 5 из которых классификатор также определил верно. Тогда accuracy:

$$accuracy = \frac{5+90}{5+90+10+5} = 0.864$$

Если мы просто будем предсказывать все письма как не-спам, то получим более высокую accuracy

$$accuracy = \frac{0+100}{0+100+0+10} = 0.909$$

Метрики качества классификации

- precision (точность) и recall (полнота).

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

Precision показывает долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

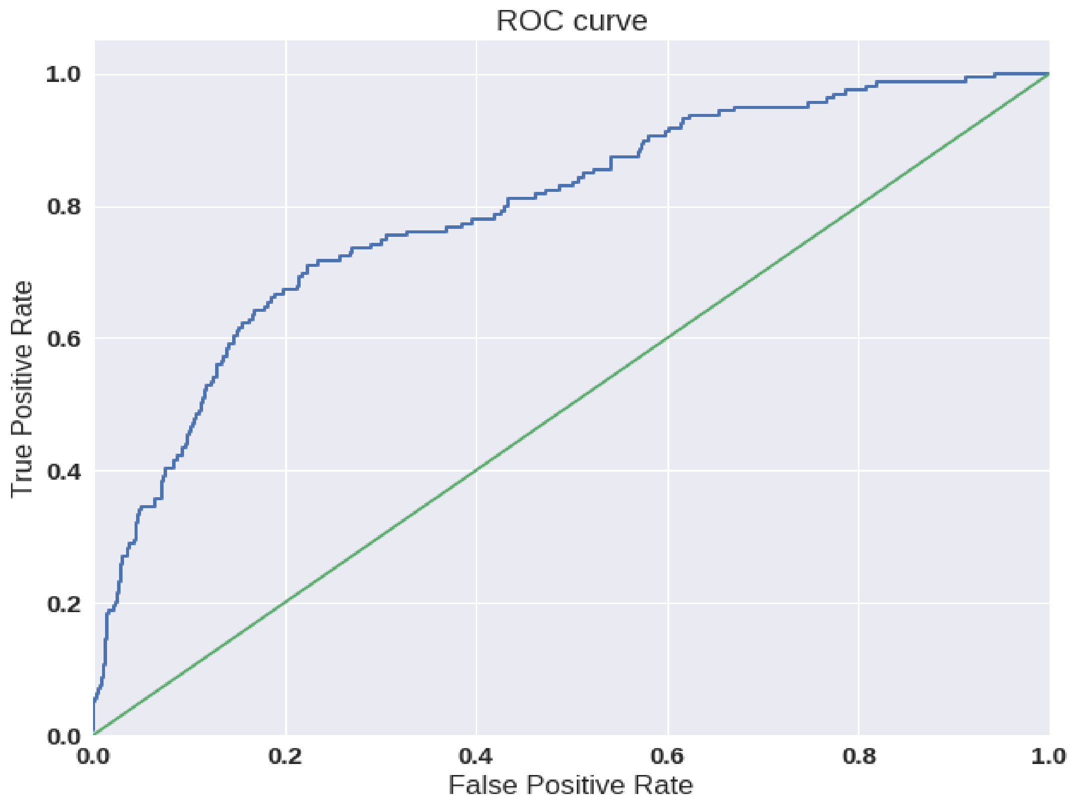
Precision не позволяет записывать все объекты в один класс, так как в этом случае растет значение FP. Recall демонстрирует способность алгоритма обнаруживать данный класс вообще, а precision — способность отличать этот класс от других классов.

AUC-ROC –площадь под кривой ошибок

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

TPR - это полнота, а FPR показывает, какую долю из объектов отрицательного класса алгоритм предсказал неверно.



Кривая ошибок или **ROC-кривая** – графичекая характеристика качества бинарного классификатора, зависимость доли верных положительных классификаций от доли ложных положительных классификаций при варьировании порога решающего правила.

AUC-ROC –площадь под кривой ошибок

В идеальном случае, когда классификатор не делает ошибок (FPR = 0, TPR = 1), площадь под кривой, равна 1; в противном случае, когда классификатор случайно выдает вероятности классов, AUC-ROC = 0.5. Каждая точка на графике соответствует выбору некоторого порога вероятности, разделяющего положительный и отрицательный класс.

Площадь под кривой в данном случае показывает качество алгоритма (больше — лучше), кроме этого, важной является крутизна самой кривой — мы хотим максимизировать TPR, минимизируя FPR, а значит, наша кривая в идеале должна стремиться к точке (0,1).

Критерий AUC-ROC устойчив к несбалансированным классам и может быть интерпретирован как вероятность того, что случайно выбранный положительный объект будет иметь более высокую вероятность быть положительно определенным данным классификатором, чем случайно выбранный отрицательный объект.

Чувствительность и специфичность

- Наряду с FPR и TPR при оценке качества классификации используют также понятия *чувствительности* и *специфичности*, которые изменяются в интервале [0,1]:
- *чувствительность* алгоритма совпадает с TPR (долей положительных объектов, правильно классифицированных алгоритмом);
- *специфичность* алгоритма определяется как 1-FPR (это доля отрицательных объектов, правильно классифицированных алгоритмом).
- Модель с высокой чувствительностью чаще дает истинный результат при наличии положительного исхода (хорошо обнаруживает положительные примеры). Наоборот, модель с высокой специфичностью чаще дает истинный результат при наличии отрицательного исхода (хорошо обнаруживает отрицательные примеры).