Data Mining in Action

Про метрики

Алексей Горчаков

2019 10 19 Суббота

Эпиграф

- “Если вы не можете что-то измерить, то вы не можете и улучшить это”
  - Кто-то приписывет это высказывание Уильяму Томсону, лорду Кельвину
  - Кто-то приписывает его отцу современного менеджмента Питеру Друкеру
  - Я бы добавил, что вы не сможете понять, что вы это улучшили.
    - Тут стоит пошутить про продуктовое видение

План

- Мы поговорим про постановку задачи оценки качества решения
- Вспомним, что даже до привлечения признакового описания можно делать прогнозы
- Мы поговрим про метрики бинрной классификации
  - Попутно еще раз быстро их вспомнив
- Мы поговрим про метрики регрессии
  - И их еще раз быстро вспомним

Наивная таксономия метрик качества

- метрики в производственных системах
  - онлайн/оффлайн
  - качество/бизнес
    - Прокси
- метрики при избретении новых методов машинного обучения
- метрики в соревнованиях по анализу данных

Общение с аудиторией

- Для чего нужны метрики в соревнованиях по машинному обучению?
  - Говоря о метриках, мы говорим о метриках качества в первую очередь.

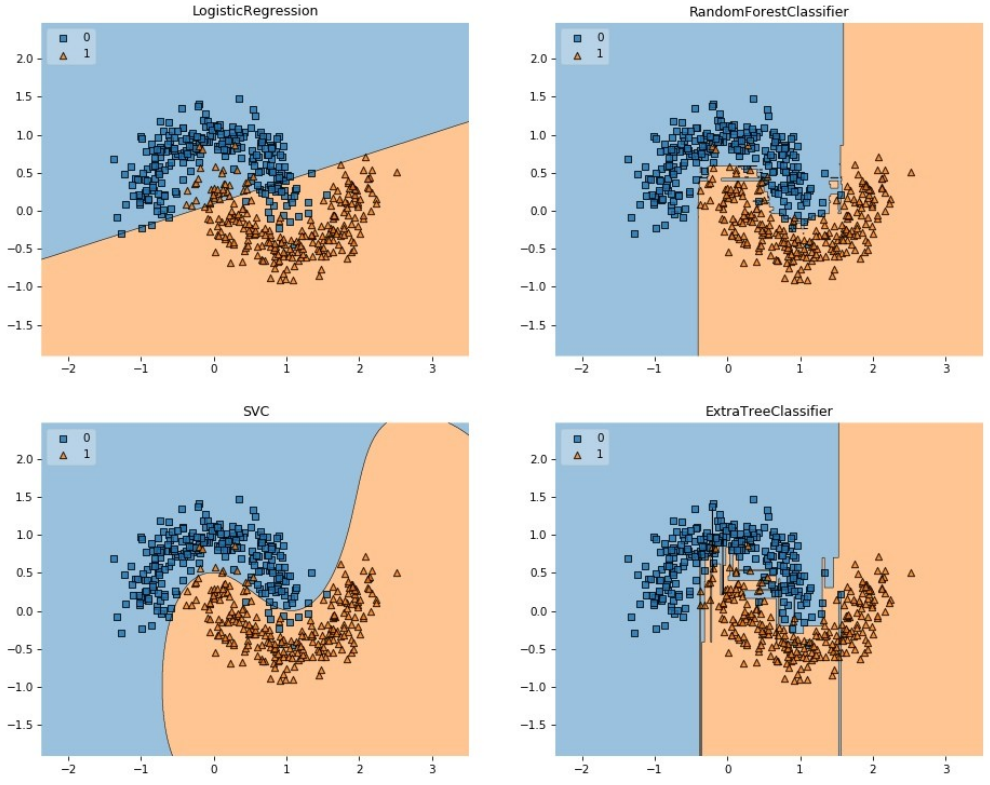
Ответ лектора

- Чтобы выбрать победителя
- Ибо организаторы надеятся, что значение метрики адекватно оценивает качество нашего решения поставленной пред нами задачи
  - Вообще говоря, оценка получается точечной, со всеми вытекающими из этого последствиями
    - при сравнении двух подходов мы можем предпочесть один другому из-за шума
    - Мораль: ищи такие методы, котрые с "зазором" победят ближайших коллег по соревнованию. Иначе готовься к shakeup

Формальная постановка задача выбора метрики

- Есть истинные данные y_true на наборе объектов (метки, вещественные числа, списки каких-то сущностей, регионы на изображении и т.п.)
- Есть предсказания y_pred, полученные по признаковому описанию техже объектов
- Задача: численно оценить близость y_true и y_pred

Эти модели решают одну задачу



Но нам доступны только сухие значения разных метрик...

LogisticRegression

	precision	recall	f1-score	support
0	0.86	0.85	0.85	254
1	0.84	0.86	0.85	246
micro avg	0.85	0.85	0.85	500
macro avg	0.85	0.85	0.85	500
weighted avg	0.85	0.85	0.85	500

RandomForestClassifier

	precision	recall	f1-score	support
0	1.00	0.99	0.99	252
1	0.99	1.00	0.99	248
micro avg	0.99	0.99	0.99	500
macro avg	0.99	0.99	0.99	500
weighted avg	0.99	0.99	0.99	500

SVC

	precision	recall	f1-score	support
0	0.96	0.96	0.96	250
1	0.96	0.96	0.96	250
micro avg	0.96	0.96	0.96	500
macro avg	0.96	0.96	0.96	500
weighted avg	0.96	0.96	0.96	500

ExtraTreeClassifier

	precision	recall	f1-score	support
0	1.00	1.00	1.00	250
1	1.00	1.00	1.00	250
micro avg	1.00	1.00	1.00	500
macro avg	1.00	1.00	1.00	500
weighted avg	1.00	1.00	1.00	500

Что человечество придумало для оценки

- Метрики соревнований с boosters.pro:
  - RMSLE
  - MAE
  - WRMSE
  - AUC ROC
  - Accuracy
  - Logloss
  - Equal Error Rate
  - MNAP @ K

Почему метрик много?

- Вообще говоря, решение нашей задачи должно удовлетворять разному набору аспектов качества
- За нас в соревновании выбор сделал организатор. По его мнению, выбранная метрика лучше всего отражает требоания к решению

Что нам остается?

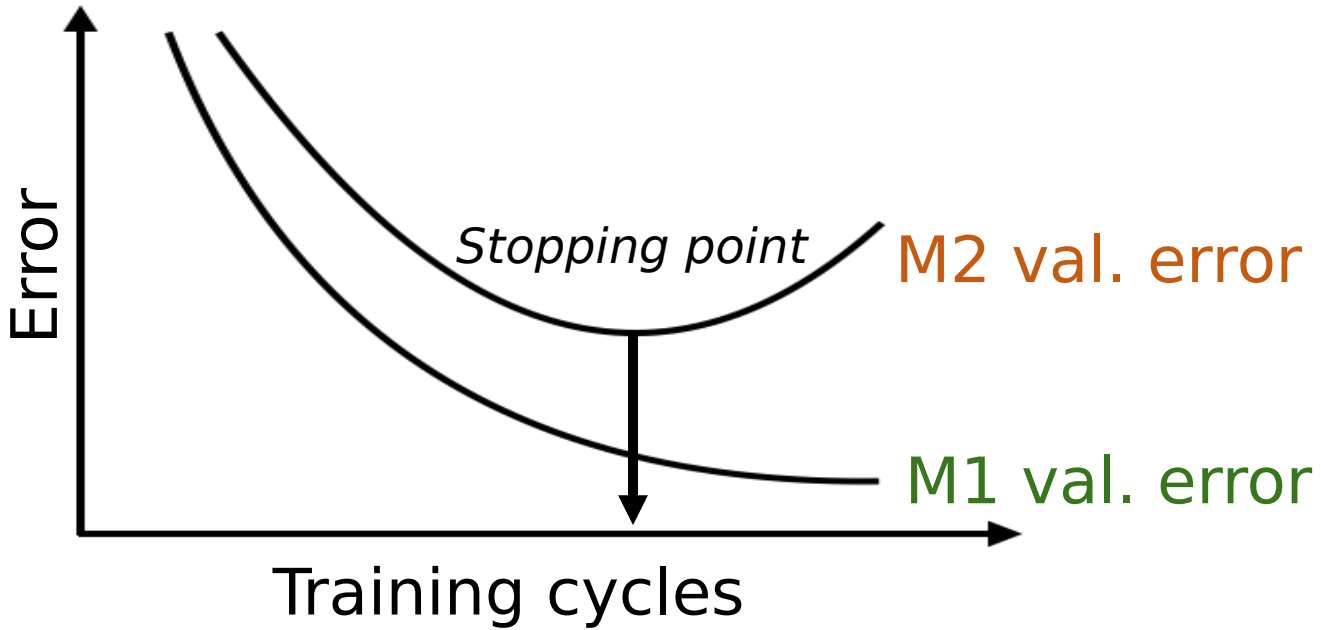
- Грамотно применить инструменты для оптимизации конкретной метрики качества в конкретном соревновании

Метрики и функции потерь

- Нужно понимать, как устроены наши инструменты: что реально они пытаются сделать (они обычно на train минимизируют функцию потерь)
- Если метрика соревнования напрямую не оптимизируется инструментами, придется подбирать гиперпараметры метода обучения так, чтобы минимизировать метрику задачи
  - Опасность: сама модель может переобучиться под под target на train в этот момент

Подходы к оптимизации target metric

- Оптимизируем метрику M1, отслеживаем метрику M2
  - Выходим когда M2 лучшая



План

- ~~Мы поговорим про постановку задачи оценки качества решения~~
- Вспомним, что даже до привлечения признакового описания можно делать прогнозы
- Мы поговрим про метрики бинрной классификации
  - Попутно еще раз быстро их вспомнив
- Мы поговрим про метрики регрессии
  - И их еще раз быстро вспомним

Константное предсказание

- Прежде чем начинать моделировать, полезно поставить точку отсчета: сделать несколько константных предсказаний и посмотреть, как они на public leaderboard себя проявляют
- Некоторые метрики качества допускают аналитическое вычисление наилучшего константного предсказания
- Но, если лень думать, то scipy.optimize
  - Вообще говоря, можно построить вероятностную модель и использовать всю мощь науки про оценку параметров распределения по ряду реализованных случайных величин.
  - Предсказывать тогда можно, например, математическим ожиданием или каким-то из квантилей

Константное предсказание

sklearn.dummy.DummyClassifier¶

class `sklearn.dummy.` **`DummyClassifier`** (*strategy='stratified', random_state=None, constant=None*) [source]

DummyClassifier is a classifier that makes predictions using simple rules.

This classifier is useful as a simple baseline to compare with other (real) classifiers. Do not use it for real problems.

Read more in the User Guide.

Parameters:

strategy : *str, default="stratified"*

Strategy to use to generate predictions.

- "stratified": generates predictions by respecting the training set's class distribution.
- "most_frequent": always predicts the most frequent label in the training set.
- "prior": always predicts the class that maximizes the class prior (like "most_frequent") and `predict_proba` returns the class prior.
- "uniform": generates predictions uniformly at random.
- "constant": always predicts a constant label that is provided by the user. This is useful for metrics that evaluate a non-majority class

New in version 0.17: Dummy Classifier now supports prior fitting strategy using parameter *prior*.

random_state : *int, RandomState instance or None, optional, default=None*

If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

constant : *int or str or array of shape = [n_outputs]*

The explicit constant as predicted by the "constant" strategy. This parameter is useful only for the "constant" strategy.

План

- ~~Мы поговорим про постановку задачи оценки качества решения~~
- ~~Вспомним, что даже до привлечения признакового описания можно делать прогнозы~~
- Мы поговрим про метрики бинрной классификации
  - Попутно еще раз быстро их вспомнив
- Мы поговрим про метрики регрессии
  - И их еще раз быстро вспомним

Метрики бинарной классификации

- Задачи могут быть нескольких типов
  - Нужно ответить какой класс у объекта
    - Какую иконку облачности/осадков нарисовать
    - Нажимать ли на тормоз самодвижемуся автомобилю
    - Выдавать ли кредит
  - Нужно оценить вероятность принадлежности к классу
    - Какова вероятность слонечной погоды/дождя
    - Какова вероятность, что пешеход выйдет перед машиной
    - Какова вероятность, что клиент не вернет кредит

Метрики качества выбора класса

- Метрики можно вычислять как функции от Confusion matrix
  - Не различающие ошибки первого и второго родов
    - (Balanced) Accuracy
  - Различающие
    - Precision
    - Recall
    - F1
      - Семейство F-measure

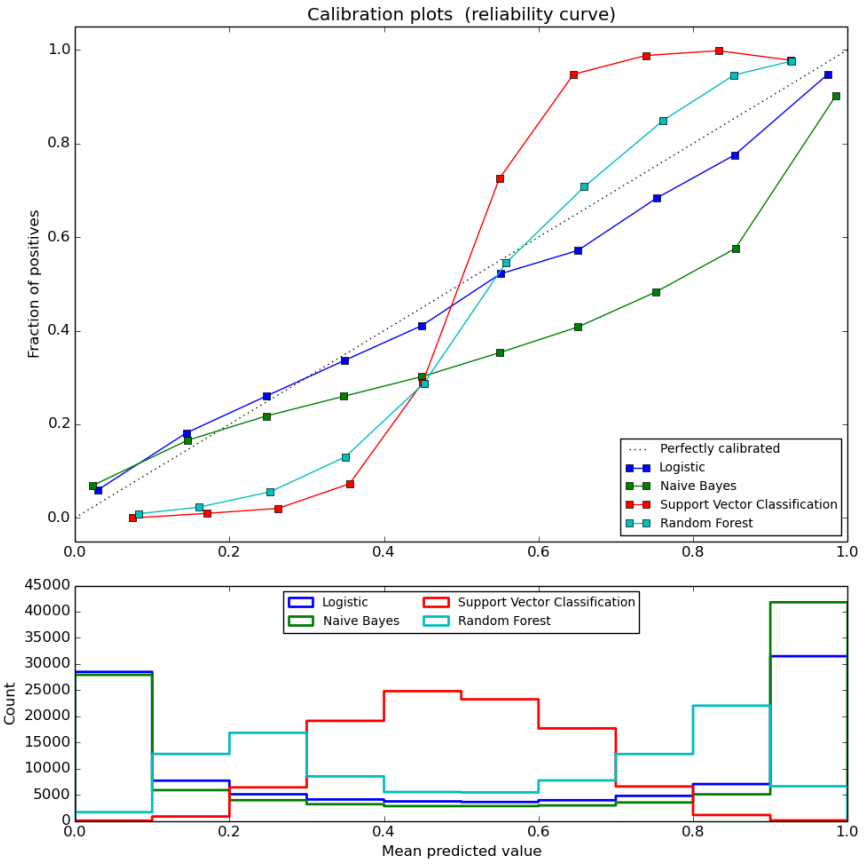
Метрики качества принадлежности к классу

- Правдоподобие или logloss
- AUC ROC

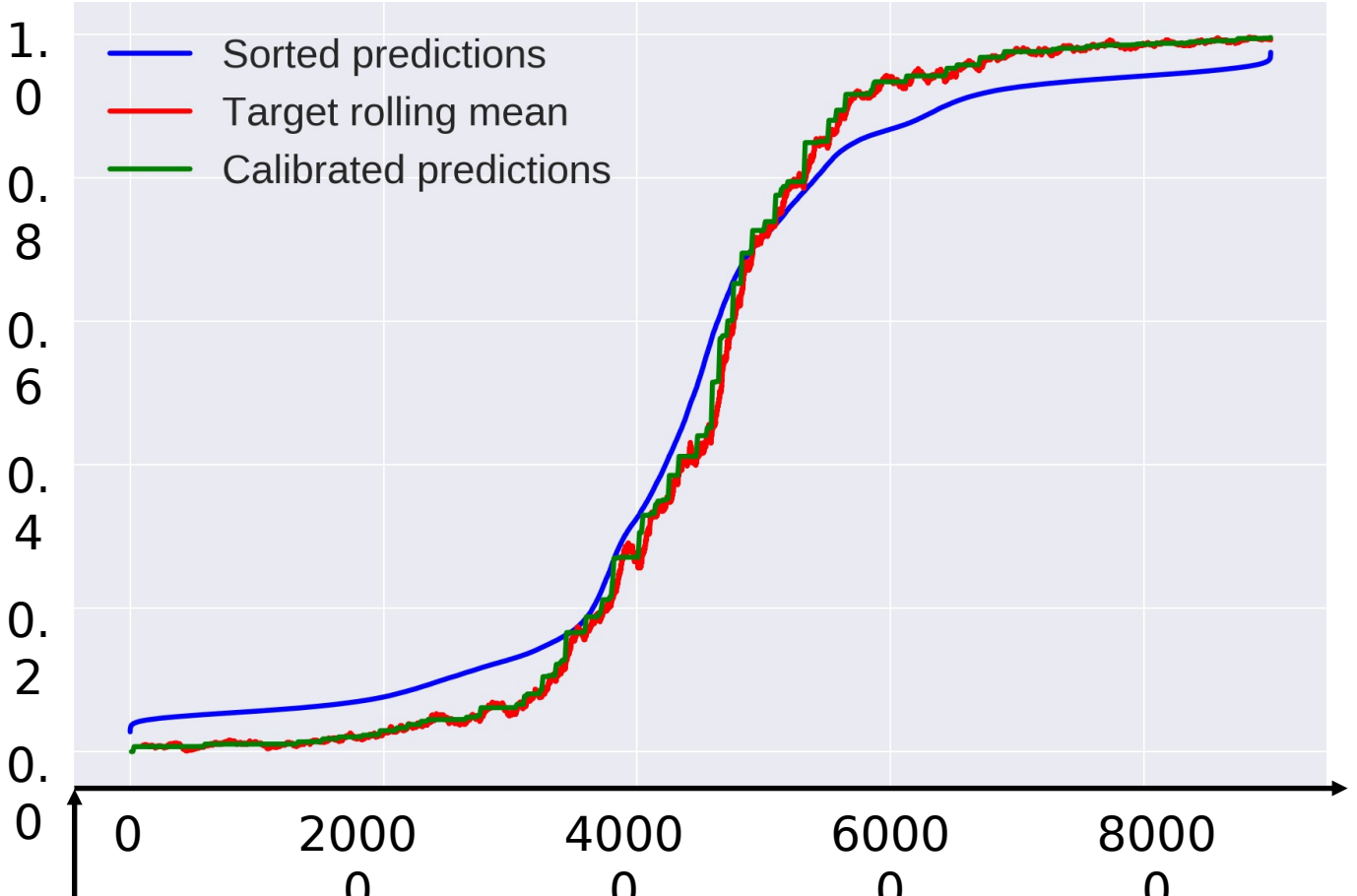
Ранжировщики и предсказатели вероятностей

- Если модель явно не оптимизирует logloss, то она может выдвать числа, которые лишь верно упорядочивают объекты по вероятности принадлежности к классу 1
  - Пример: KNNClassifier, RandomForestClassifier
- Если это так, то необходимо воспользоваться процедурами калибровки

Ранжировщики и предсказатели вероятностей



Калибровка вероятностей



Калибровка вероятностей

- Platt scaling
  - Просто обучи Logistic Regression на свои предсказания (похоже на стакинг)
- Isotonic regression
  - Просто обучи Isotonic Regression на свои предсказания (похоже на стакинг)
- Stacking
  - Просто обучи XGBoost или neural net на свои предсказания (точно стакинг)

Непосредственно метрики

- Дальше быстро пробежимся по метрикам

Confusion matrix

Пример:

```
In [14]: # Важно: первый аргумент - true values, второй - predicted values
         # получаем матрицу 2x2)
         print(metrics.confusion_matrix(y_test, y_pred_class))
```

```
[[118  12]
 [ 47  15]]
```

N = 192	Предсказан: **0**	Предсказана: **1**
Истинный: **0**	118	12
Истинный: **1**	47	15

TP, FP, TN, FN

N = 192	Предсказан: **0**	Предсказана: **1**
Истинный: **0**	TN = 118	FP =12
Истинный: **1**	FN = 47	TP = 15

- True Positives (TP): мы правильно предсказали 1
  - 15
- True Negatives (TN): мы правильно предсказали 0
  - 118
- False Positives (FP): мы неправильно предсказали 1
  - 12
  - Ошибка 1 типа
- False Negatives (FN): мы неправильно предсказали 0
  - 47
  - Ошибка 2 типа

Метрики из Confusion matrix

- Accuracy: $\frac{(TP+TN)}{(TP+TN+FP+FN)} = 0.693$

- Sensitivity:
  - Когда исходное значение позитивны(1), как часто предсказания верны?
  - Как "sensitive" классификатор к обнаружению положительных классу?
  - Также известен как "True Positive Rate (TPR)" или "Recall“ $\frac{TP}{(TN+FN)} = 0.242$
- Specificity:
  - Как "specific" (или "selective") классификатор в предсказании позитивного класса?
  - False Positive Rate (FPR) = (1 – Specificity) $\frac{TN}{(TN+FP)} = 0.907$

Метрики из Confusion matrix

- Prevalence: $\frac{(TN+FN)}{(TN+FP+FN+FP)} = 0.859$
  - Какой процент позитивных(1) значений у нас?
  - Более высокая Prevalence подразумевает, что вы можете получить более высокую Precision даже при угадывании.
- Detection Rate: $\frac{TN}{(TN+FP+FN+FP)} = 0.615$
  - Правильно предсказанные 1 во всем сете
  - Если у нас хорошая точность модели, при среднем Detection Rate, как будет меняться точность если в выборке будет больше единиц.

Метрики из Confusion matrix

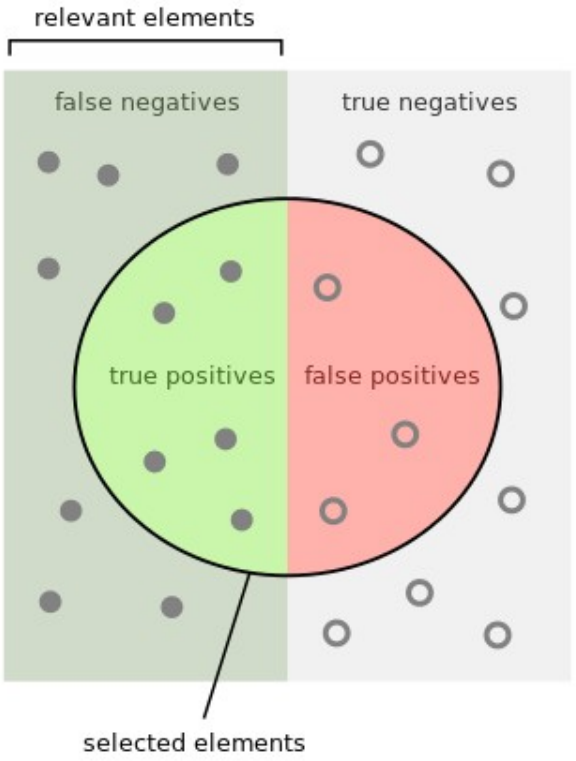
- Balanced Accuracy: $\frac{Sensitivity+Specificity}{2} = 0.574$
  - Процент правильных предсказаний среди 0 и 1
  - Подходит, если ваша выборка несбалансированная
- Precision: $\frac{TP}{(TP+FP)} = 0.556$
  - Когда мы предсказываем 1, как часто это верно
- F1-score: $\frac{2*Precision*Recall}{(Precision+Recall)} = 0.337$
  - Это комбинация Precision и Recall
  - Подходит, если есть дисбаланс в выборке

Accuracy

```
Accuracy = np.mean(ytrue == ypred)
```

Лучшее константное решение - самый часто встречающийся класс

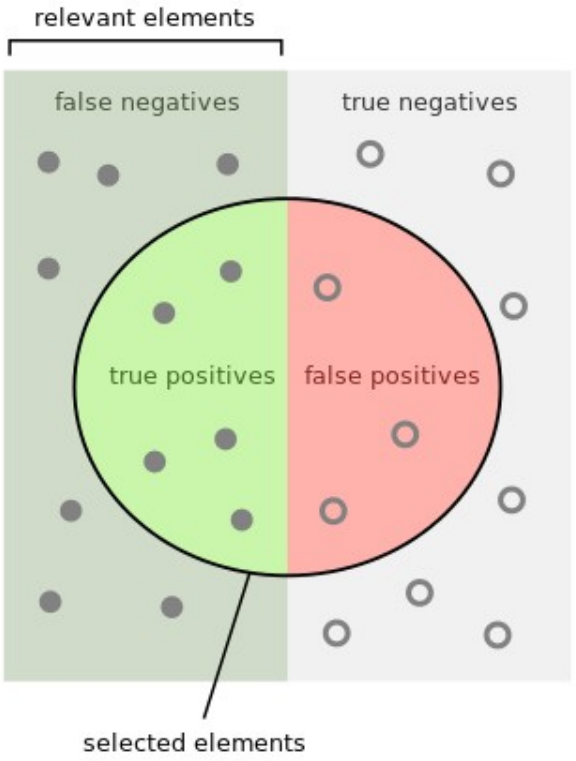
Precision and recall

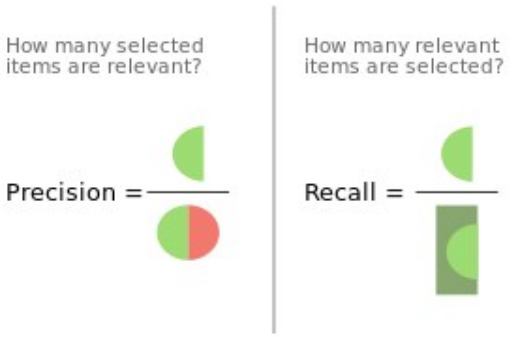


False Positive - ошибка I рода (ложное срабатывание)

False Negative - ошибка II рода (объект пропущен)

Precision and recall



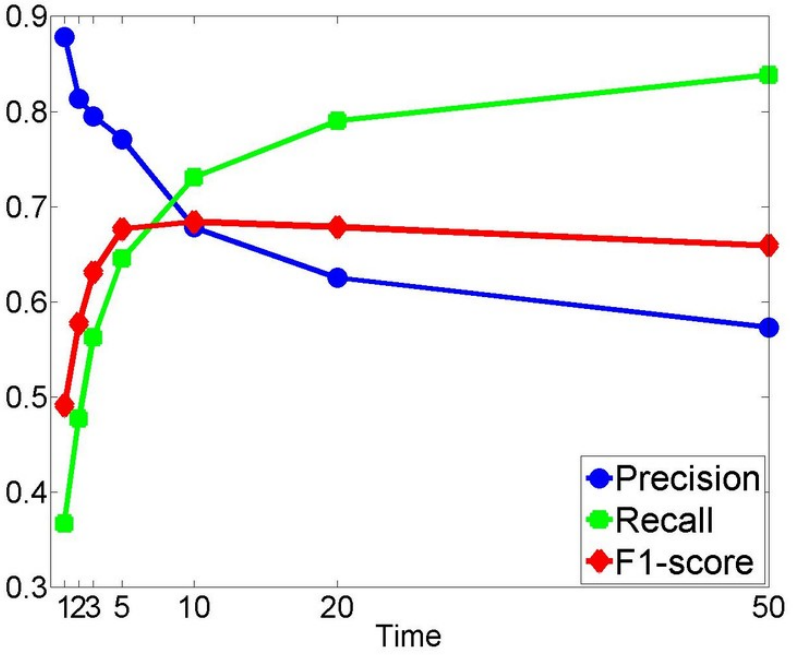


$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

F-score

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



F-score

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

```
y_true = [[1, 2],
          [3, 4, 5],
          [6],
          [7]]
y_pred = [[1, 2, 3, 9],
          [3, 4],
          [6, 12],
          [1]]

mean_f1(y_true, y_pred)
# 0.53333333
```

F-score

$$F_\beta = \left(1+\beta^2\right) * \frac{\text{precision} * \text{recall}}{\left(\beta^2 * \text{precision}\right) + \text{recall}}$$

при 0<beta<1 предпочтение отдаётся точности
при beta >1 больший вес приобретает полнота

AUC (ROC)

Доля правильно отранжированных пар:

```
ypred_i > ypred_j IF ytrue_i > ytrue_j
```

То же самое:

```
ypred_i > ypred_j IF ytrue_i == 1 and ytrue_j == 0
```

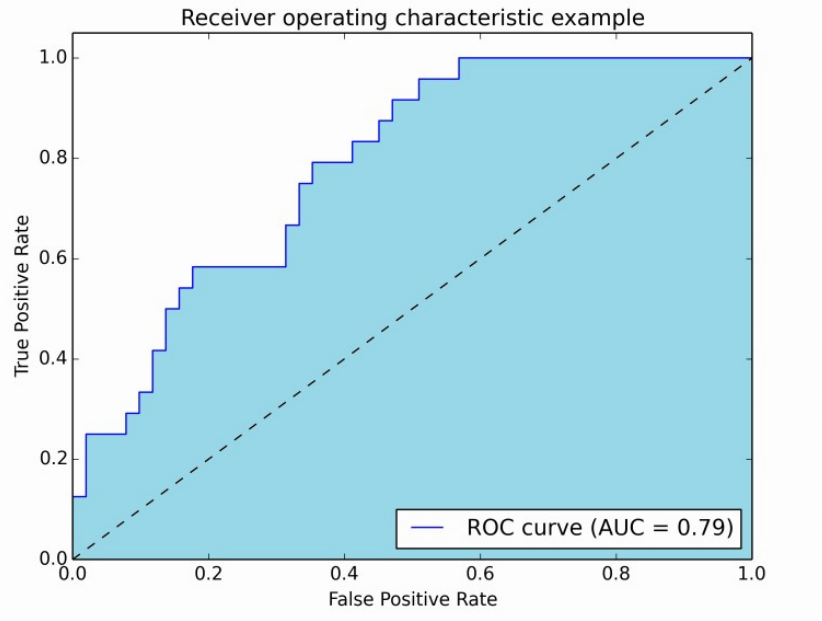
Если `ypred_i = ypred_j` считаем,

AUC (ROC)

Доля правильно отранжированных пар:

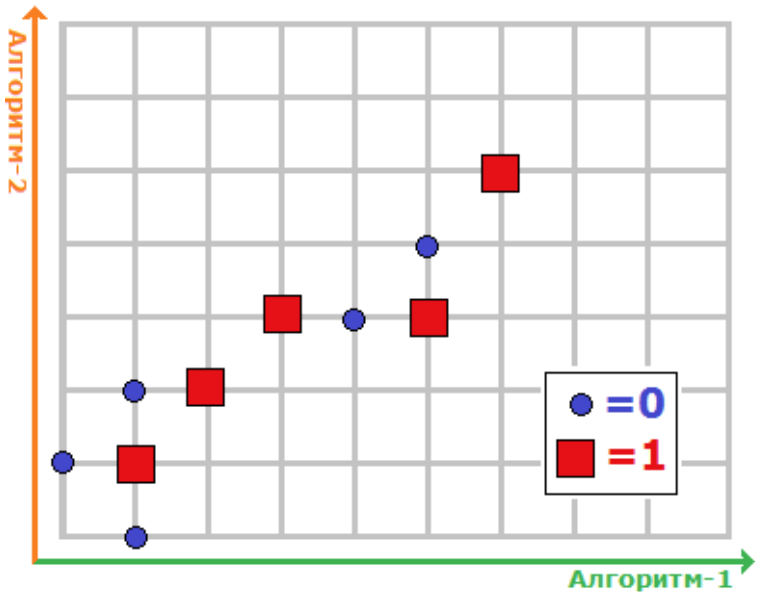
`ypred_i > ypred_j если ytrue_i > ytrue_j`

Или площадь под кривой:



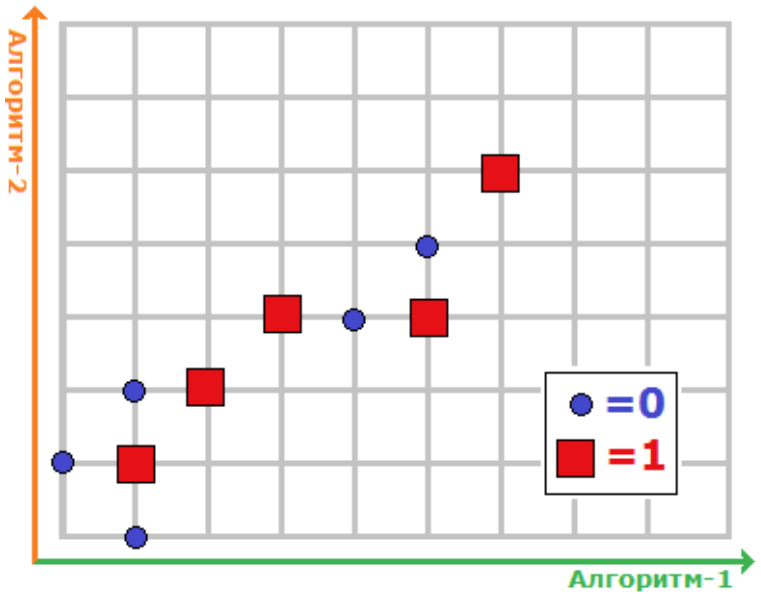
AUC (ROC)

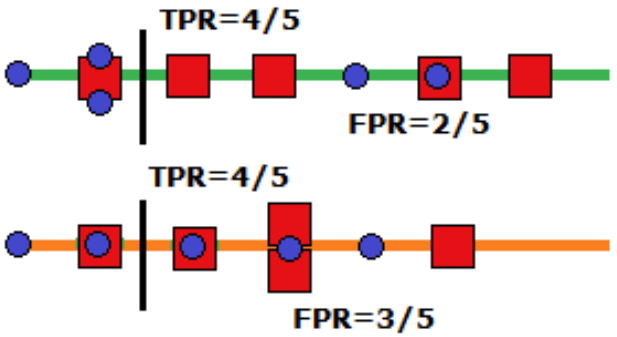
Пример: построить ROC-кривую для предсказаний двух алгоритмов



AUC (ROC)

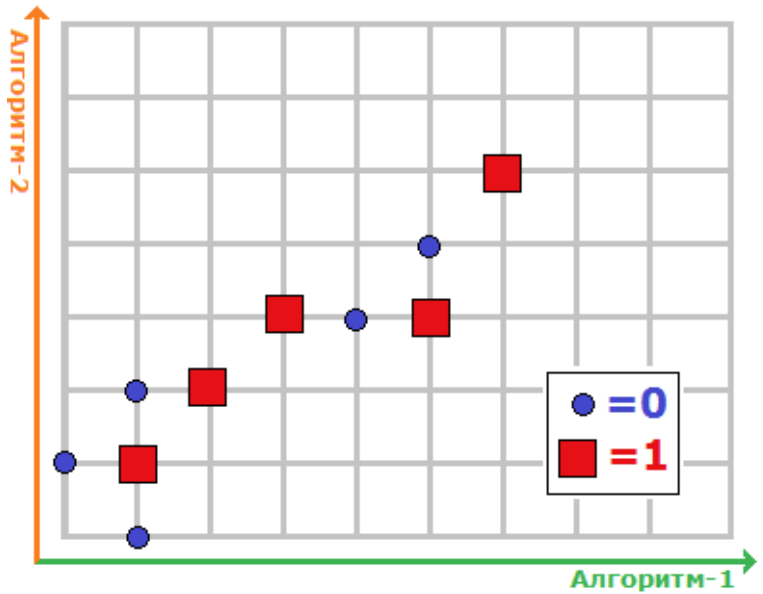
Пример: построить ROC-кривую для предсказаний двух алгоритмов

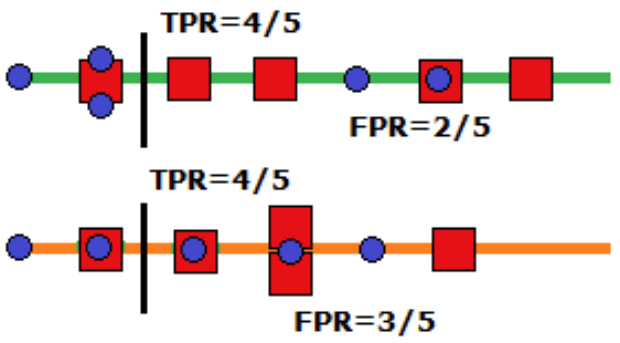


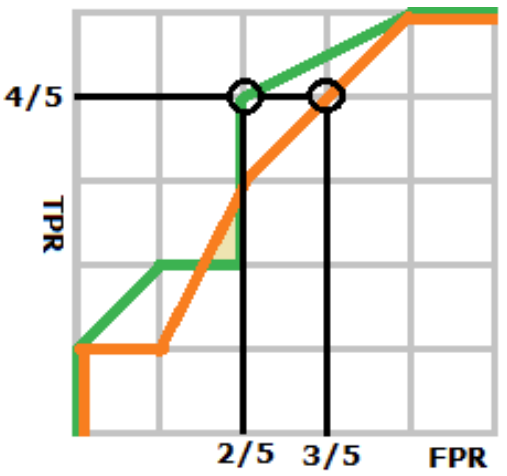


AUC (ROC)

Пример: построить ROC-кривую для предсказаний двух алгоритмов

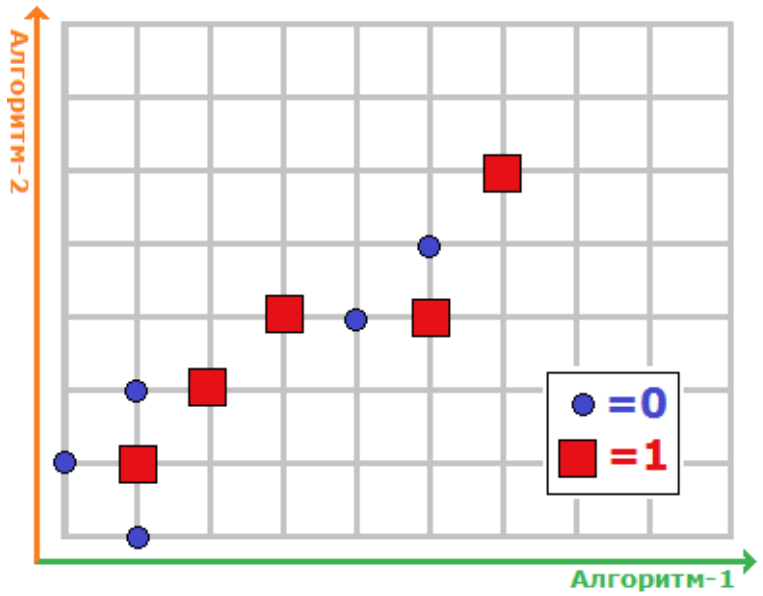


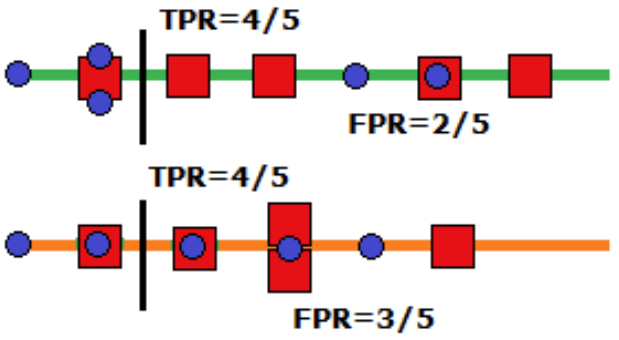


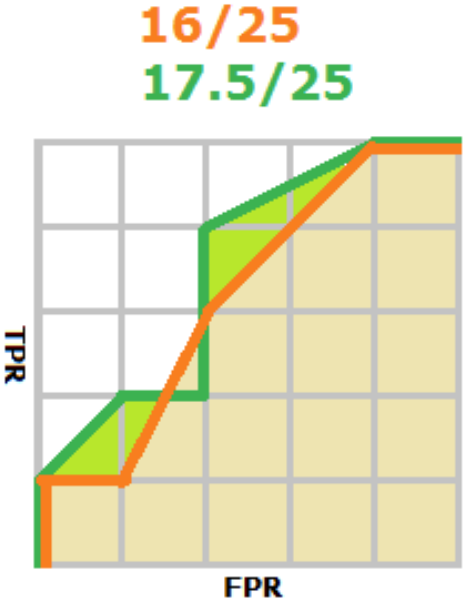


AUC (ROC)

Пример: построить ROC-кривую для предсказаний двух алгоритмов







Logloss

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

Прогноз - действительное число от 0 до 1

Лучшее константное предсказание - среднее,
то есть частота класса 1

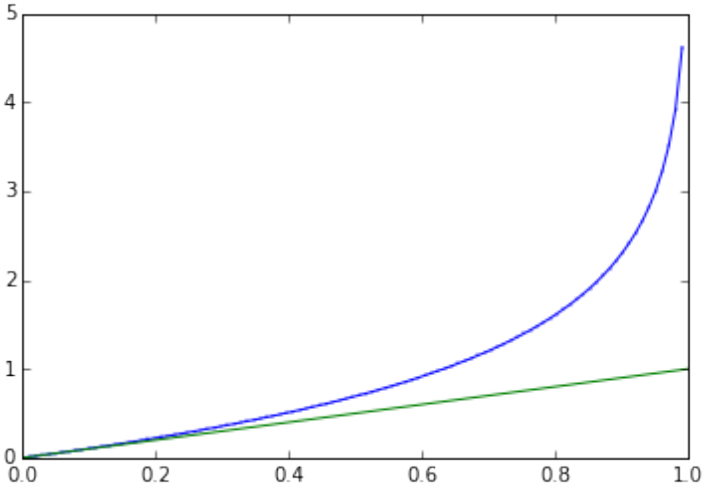
Logloss

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

Прогноз - действительное число от 0 до 1

Лучшее константное предсказание - среднее,
то есть частота класса 1

Выгодней сделать много незначительно отличающихся от истины предсказаний, чем мало, отличающихся значительно



По X: abs(ytrue - ypred)

По У: LogLoss

План

- ~~Мы поговорим про постановку задачи оценки качества решения~~
- ~~Вспомним, что даже до привлечения признакового описания можно делать прогнозы~~
- ~~Мы поговрим про метрики бинрной классификкации~~
  - ~~Попутно еще раз быстро их вспомнив~~
- Мы поговрим про метрики регрессии
  - И их еще раз быстро вспомним

Метрики регресии

- Метрики, штрафующие за абсолюнтую ошибку
  - MAE, (R)MSE
- Метрики штрафующие за относительную ошибку
  - MAPE, SMAPE, RMSLE

Mean Absolute Error

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Лучшее константное предсказание - медиана

Mean Absolute Error

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Лучшее константное предсказание - медиана

Данные:

X	Y
-1	0
-1	1
-1	1

Mean Absolute Error

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Лучшее константное предсказание - медиана

Данные:

X	Y
-1	0
-1	1
-1	1

Минимизация ошибки:

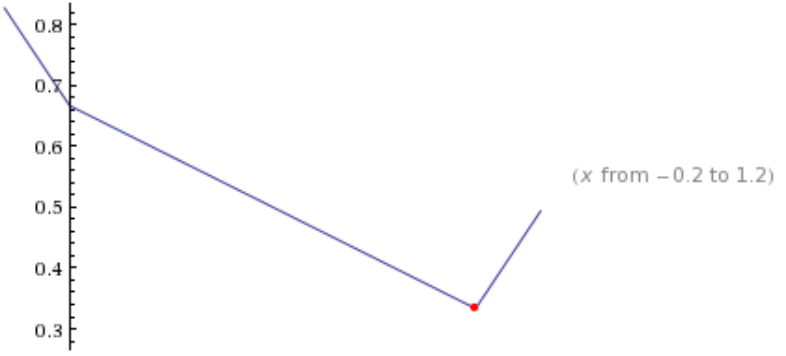
Input interpretation:

minimize	function	$\frac{1}{3}(\lvert 0-x\rvert + \lvert 1-x\rvert + \lvert 1-x\rvert)$
	domain	$0 \le x \le 1$

Global minimum:

$$\min\left\{\frac{1}{3}(|0-x|+|1-x|+|1-x|)\,\middle|\,0\le x\le 1\right\} = \frac{1}{3} \text{ at } x = 1$$

Plot:



Root Mean Squared Error

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Лучшее константное предсказание - среднее

Root Mean Squared Error

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Лучшее константное предсказание - среднее

Данные:

X	Y
-1	0
-1	1
-1	1

Минимизация ошибки:

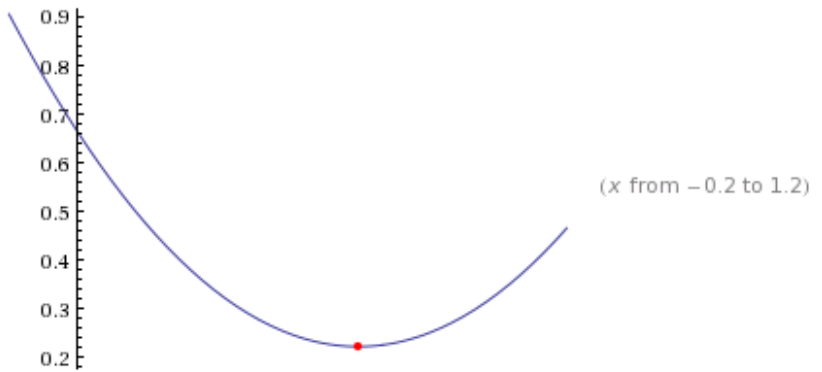
Input interpretation:

minimize	function	$\frac{1}{3}\left((0-x)^2+(1-x)^2+(1-x)^2\right)$
	domain	$0 \le x \le 1$

Global minimum:

$$\min\left\{\frac{1}{3}\left((0-x)^2+(1-x)^2+(1-x)^2\right) \,\middle|\, 0 \le x \le 1\right\} = \frac{2}{9} \text{ at } x = \frac{2}{3}$$

Plot:



Из MSE и MAE в MSPE и MAPE

- Магазин 1: предсказано 9, продано 10, MSE = 1
- Магазин 2: предсказано 999, продано 1000, MSE = 1

- Магазин 1: предсказано 9, продано 10, MSE = 1
- Магазин 2: предсказано 900, продано 1000, MSE = 1000

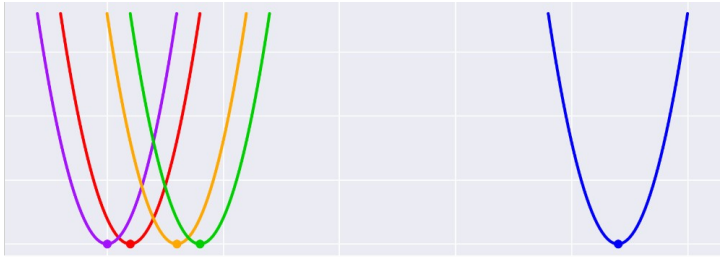
- Магазин 1: предсказано 9, продано 10, relative_metric = 1
- Магазин 2: предсказано 900, продано 1000, relative_metric = 1

* MAPE - Mean absolute percentage error

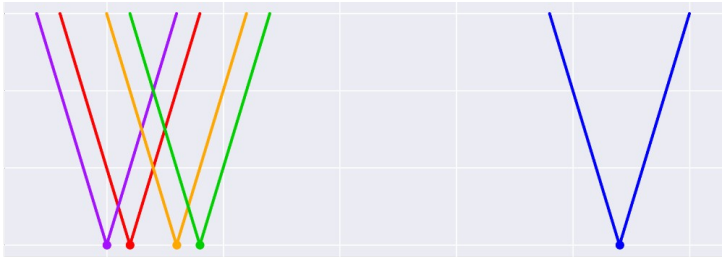
* MSPE - Mean squared prediction error

Из MSE и MAE в MSPE и MAPE

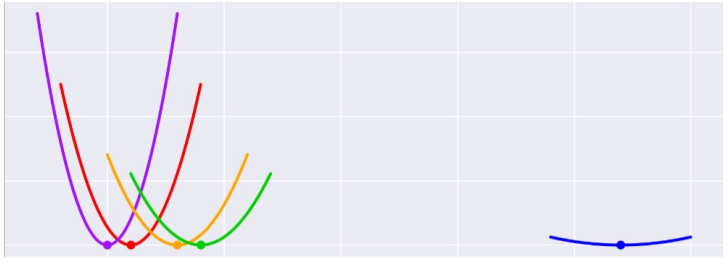
$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$



$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$



$$\mathrm{MSPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2$$



$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

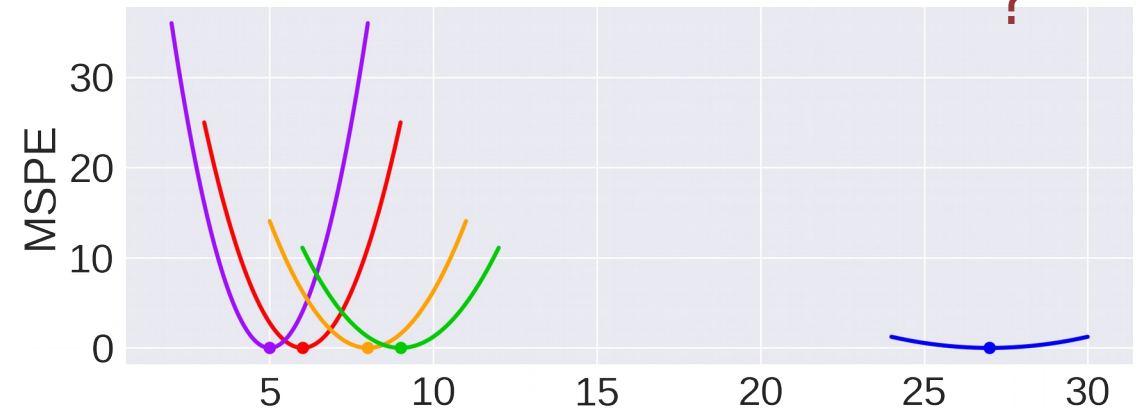
MSPE: константа

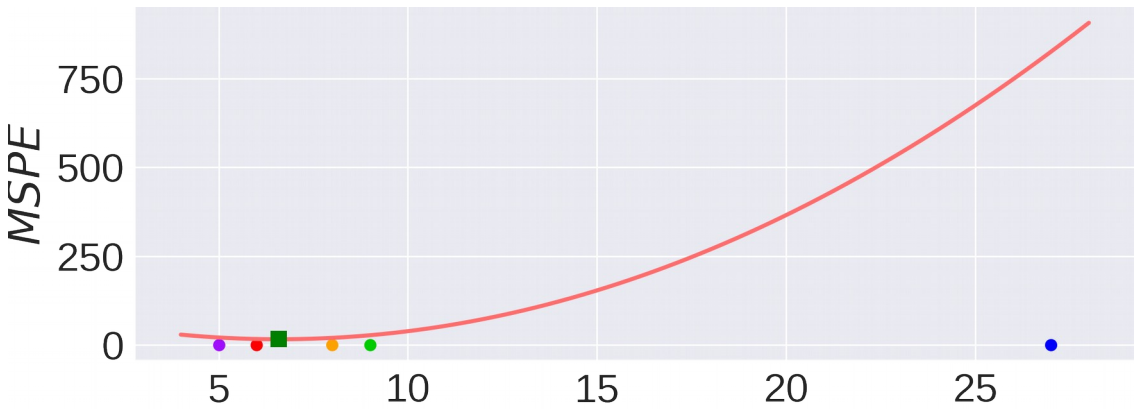
$$\mathrm{MSPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left(\frac{y_i - \alpha}{y_i}\right)^2$$

Лучшая константа: ?

Dat

X	Y
-1	4
1	3
-2	6
3	7
3	25





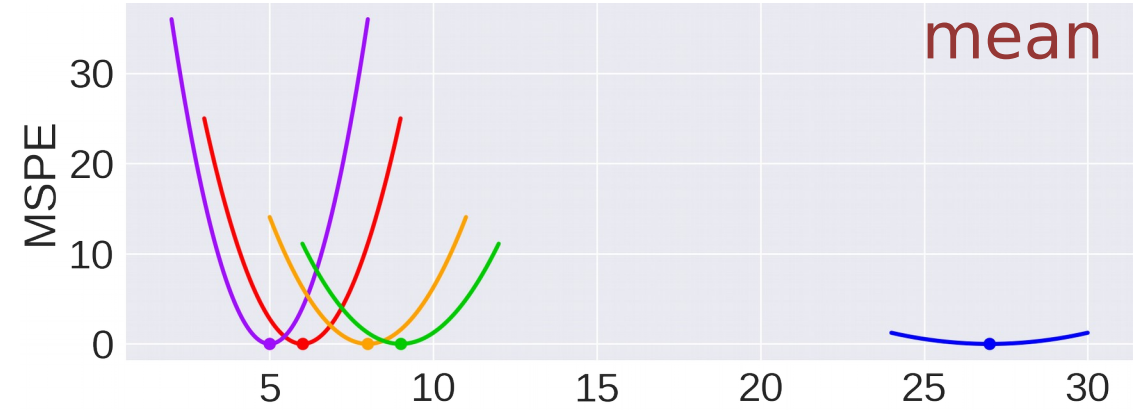
MSPE: константа

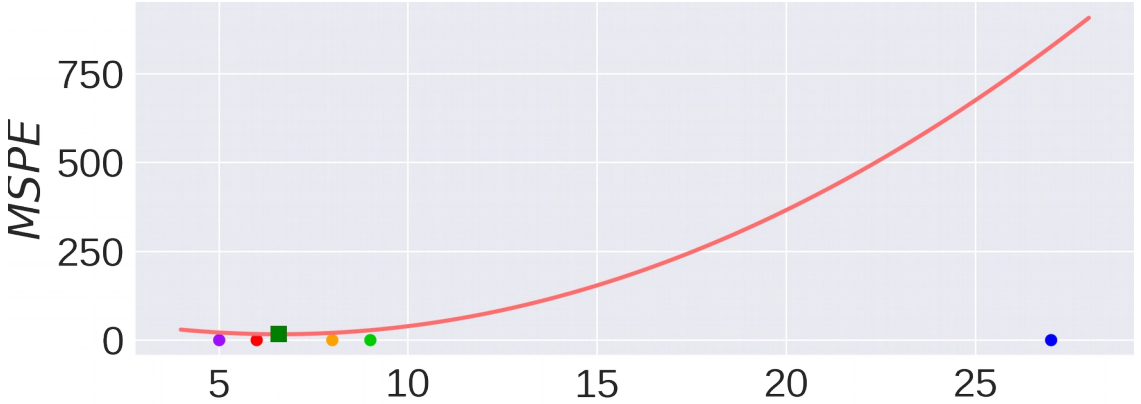
$$\text{MSPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left(\frac{y_i - \alpha}{y_i}\right)^2$$

Лучшая константа: weighted target mean

Dat

X	Y
-1	4
1	3
-2	6
3	7
3	25





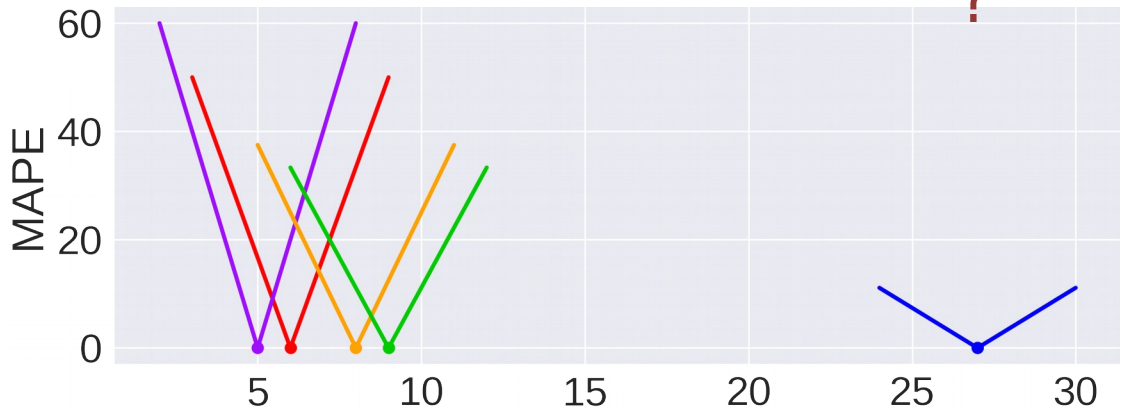
MAPE: константа

$$\mathrm{MAPE} = \frac{100\%}{N} \sum_{i=1}^{N} \left| \frac{y_i - \alpha}{y_i} \right|$$

Лучшая константа: ?

Dat

X	Y
-1	4
1	3
-2	6
3	7
3	25



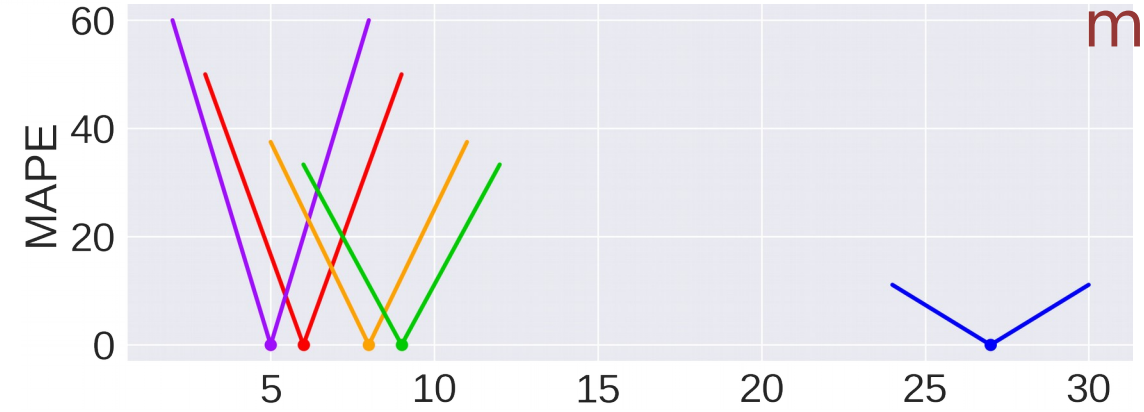
MAPE: константа

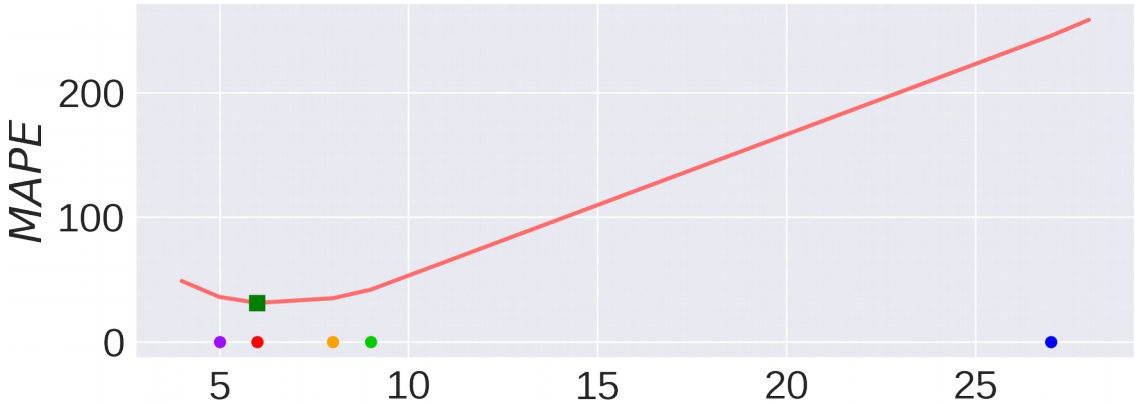
$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \alpha}{y_i}\right|$$

Лучшая константа: weighted target median

Dat

X	Y
-1	4
1	3
-2	6
3	7
3	25





(R)MSLE: Root Mean Square Logarithmic Error

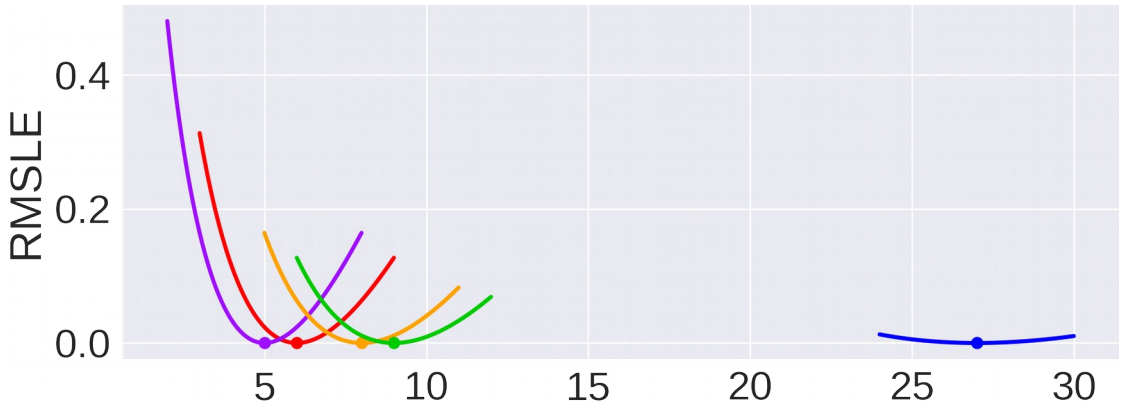
$$\begin{aligned}\text{RMSLE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\log(y_i+1)-\log(\hat{y}_i+1)\right)^2} = \\ &= RMSE\left(\log(y_i+1),\log(\hat{y}_i+1)\right) = \\ &= \sqrt{MSE\left(\log(y_i+1),\log(\hat{y}_i+1)\right)}\end{aligned}$$

(R)MSLE: Root Mean Square Logarithmic Error

Dat

X	Y
-1	4
1	3
-2	6
3	7
3	25

$$\text{RMSLE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\log(y_i+1) - \log(\hat{y}_i+1)\right)^2}$$



(R)MSLE: константа

$$\text{RMSLE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\log(y_i+1)-\log(\alpha+1)\right)^2} =$$
$$= RMSE\left(\log(y_i+1), \log(\alpha+1)\right) =$$
$$= \sqrt{MSE\left(\log(y_i+1), \log(\alpha+1)\right)}$$

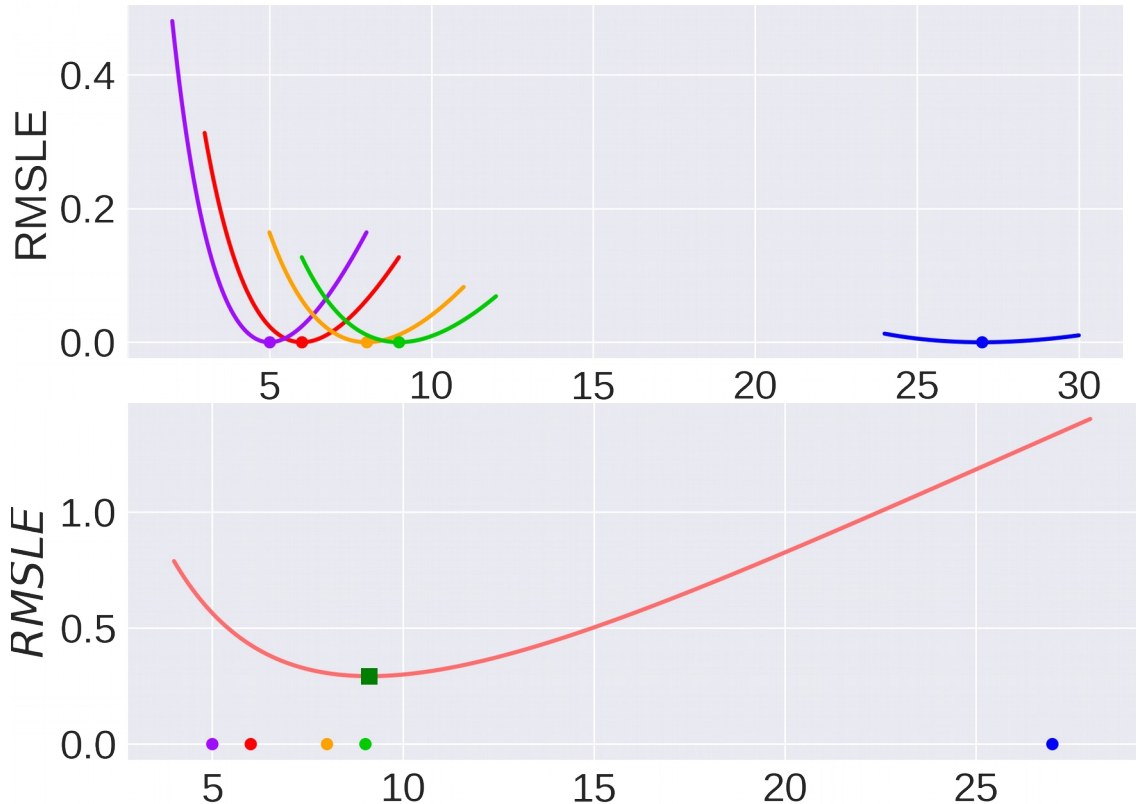
- Лучшая константа в логарифмическом пространстве это mean target value
- Чтобы получить ответ нужно взять экспоненту

(R)MSLE: константа

$$\mathrm{RMSLE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\log(y_i+1) - \log(\alpha+1)\right)^2}$$

Dat

X	Y
-1	4
1	3
-2	6
3	7
3	25



План

- ~~Мы поговорим про постановку задачи оценки качества решения~~
- ~~Вспомним, что даже до привлечения признакового описания можно делать прогнозы~~
- ~~Мы поговрим про метрики бинрной классификкации~~
  - ~~Попутно еще раз быстро их вспомнив~~
- ~~Мы поговрим про метрики регрессии~~
  - ~~И их еще раз быстро вспомним~~

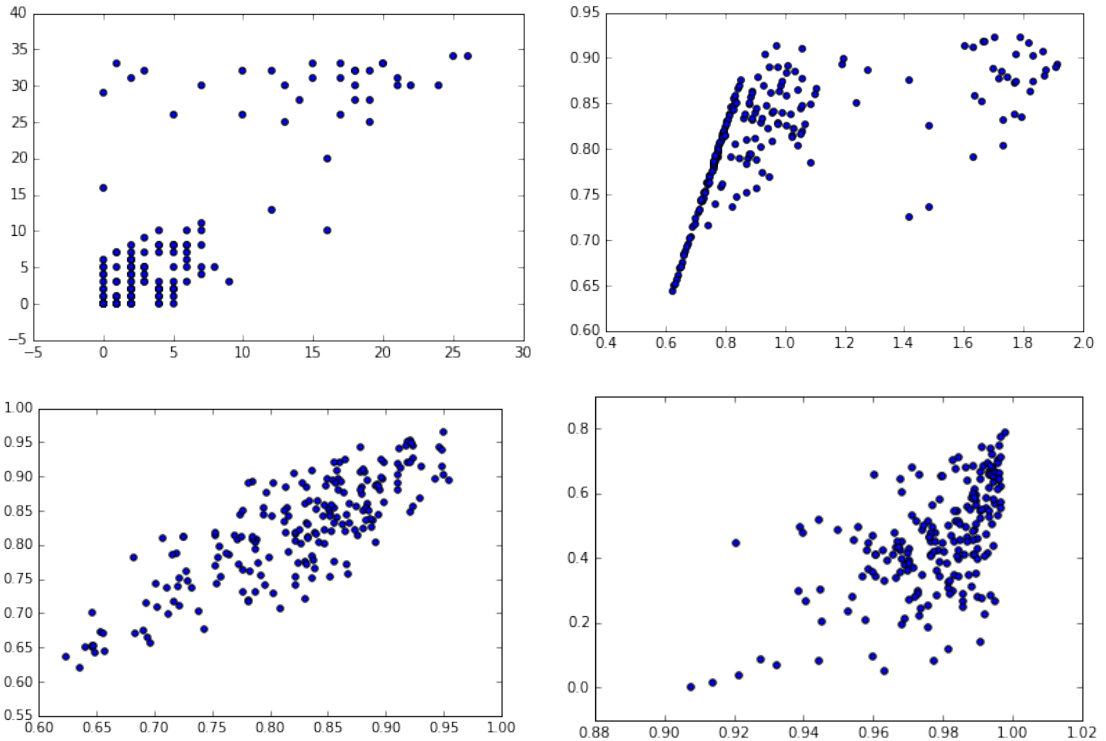
Бонус: как смешивать модели

Как смешивать?

1. Средние: арифметическое, геометрическое, гармоническое...
   Взвешивайте модели

2. Смешивание ранков

```
from scipy.stats import rankdata
ypred = rankdata(ypred1) * w1
      + rankdata(ypred2) * w2
```



Как смешивать?

1. Метрики, чувствительные к значению:
   RMSE, Logloss, etc
   y = y1 * w1 + y2 * (1 - w1) # взвешенное среднее
   y = np.sqrt(y1 * y2) # геометрическое средние

2. Метрики, чувствительные к порядку:
   AUC (ROC)
   y = y1 ** 2 + 0.3 * y2 ** 0.5 # всё, что угодно

Ссылки

- https://www.coursera.org/learn/competitive-data-science/home/week/3
- https://dyakonov.org/map/ [Метрики качества]
- https://www.youtube.com/watch?v=ypem9burgdA DMIA 2017 года