

Data Mining in Action

Лекция 4

Метрики и их оптимизация

Гущин Александр

Какие метрики бывают

1) Регрессия

- a) Mae, Mape
- b) Rmse, Rmsle

2) Классификация:

- a) Precision, Recall
- b) F-score
- c) AUC (ROC)
- d) Logloss

3) Кластеризация

(обсудили на прошлом занятии)

Mean Absolute Error

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

Лучшее константное предсказание - медиана

Mean Absolute Error

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

Лучшее константное предсказание - медиана

Данные:

X	Y
-1	0
-1	1
-1	1

Mean Absolute Error

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

Лучшее константное предсказание - медиана

Данные:

X	Y
-1	0
-1	1
-1	1

Минимизация ошибки:

Input interpretation:

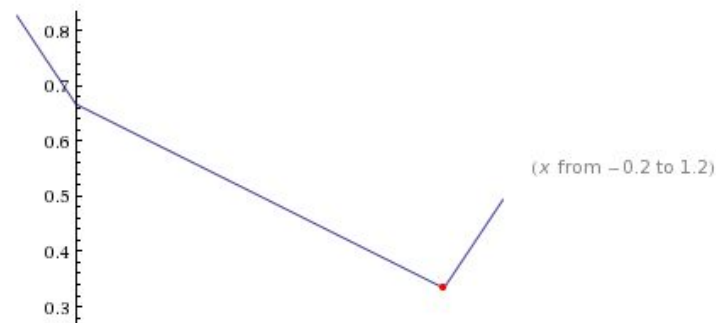
minimize	function	$\frac{1}{3} (0 - x + 1 - x + 1 - x)$
	domain	$0 \leq x \leq 1$

|2|

Global minimum:

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

Plot:



Root Mean Squared Error

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

Лучшее константное предсказание - среднее

Root Mean Squared Error

Минимизация ошибки:

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

Лучшее константное предсказание - среднее

Данные:

X	Y
-1	0
-1	1
-1	1

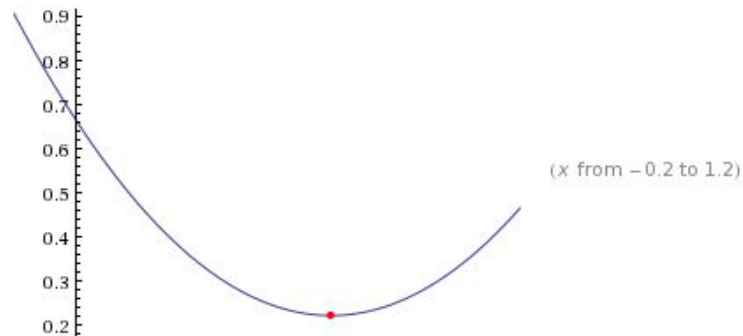
Input interpretation:

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

Global minimum:

$$\min \left\{ \frac{1}{3} ((0-x)^2 + (1-x)^2 + (1-x)^2) \mid 0 \leq x \leq 1 \right\} = \frac{2}{9} \text{ at } x = \frac{2}{3}$$

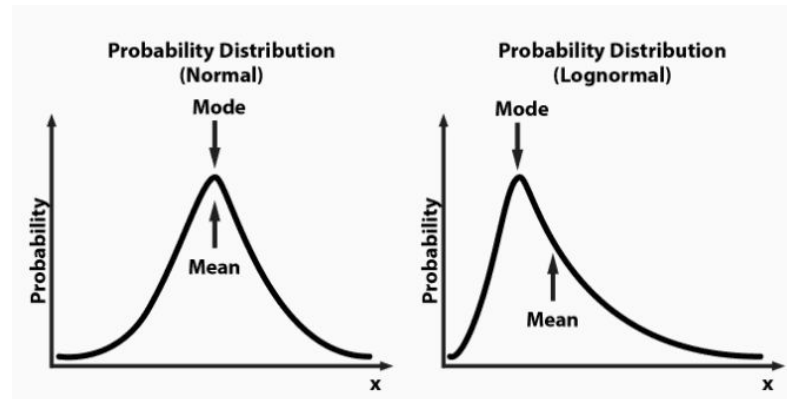
Plot:



Root Mean Squared Logarithmic Error

Лучшее константное предсказание - среднее от $\log(1 + y)$

$$\epsilon = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(p_i + 1) - \log(a_i + 1))^2}$$



Accuracy

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

Лучшее константное решение - самый часто встречающийся класс

Accuracy

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

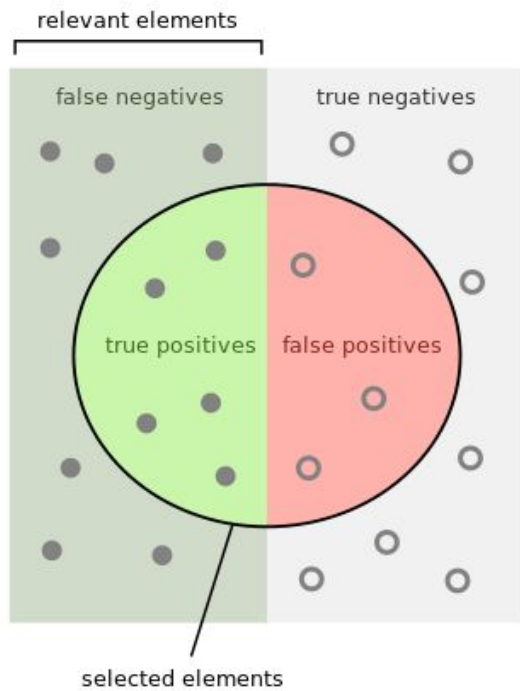
Лучшее константное решение - самый часто встречающийся класс

Пример с несбалансированными классами:

```
y={0,1}; np.mean(y) == 0.99
```

```
Accuracy = np.mean(ytrue == 1) = 0.99
```

Precision and recall

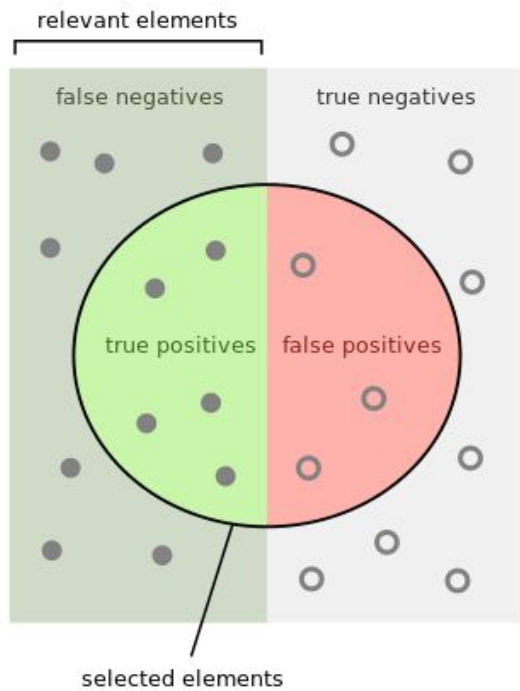


False Positive - ошибка I рода (ложное срабатывание)



False Negative - ошибка II рода (объект пропущен)

Precision and recall

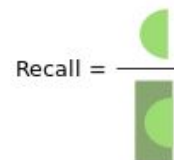


How many selected
items are relevant?



$$\text{Precision} = \frac{\text{green}}{\text{green} + \text{red}}$$

How many relevant
items are selected?



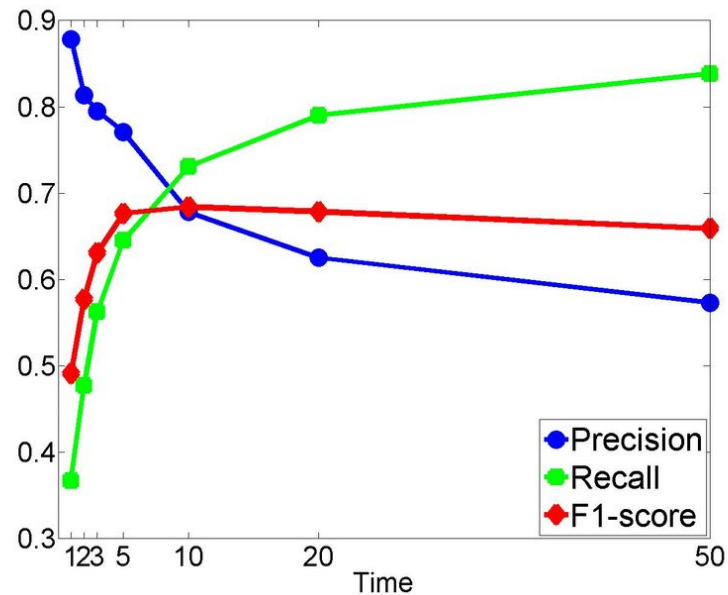
$$\text{Recall} = \frac{\text{green}}{\text{green} + \text{red}}$$

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

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

F-score

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



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

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

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



AUC (ROC)

Доля правильно отранжированных пар:

$y_{\text{pred}_i} > y_{\text{pred}_j} \text{ IF } y_{\text{true}_i} > y_{\text{true}_j}$

То же самое:

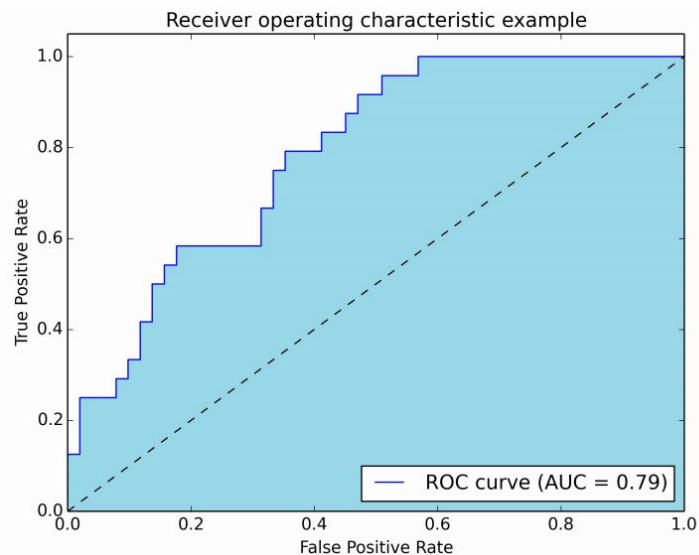
$y_{\text{pred}_i} > y_{\text{pred}_j} \text{ IF } y_{\text{true}_i} == 1 \text{ and } y_{\text{true}_j} == 0$

AUC (ROC)

Доля правильно отранжированных пар:

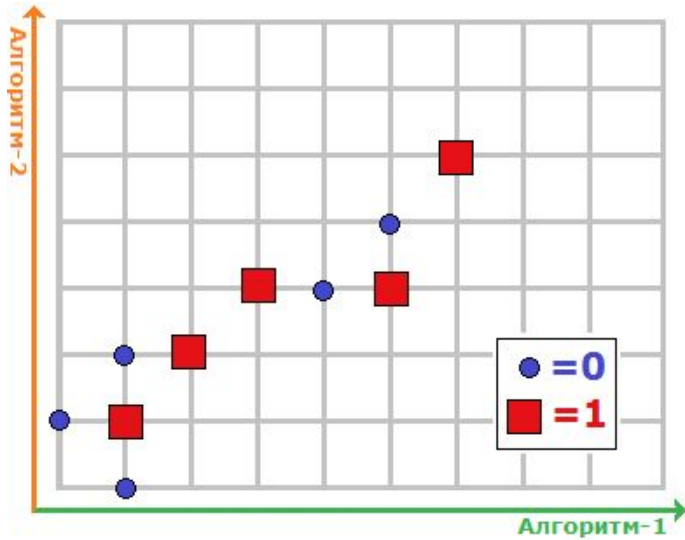
$y_{\text{pred}_i} > y_{\text{pred}_j}$ если $y_{\text{true}_i} > y_{\text{true}_j}$

Или площадь под кривой:



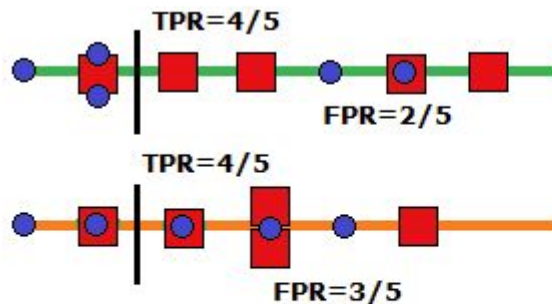
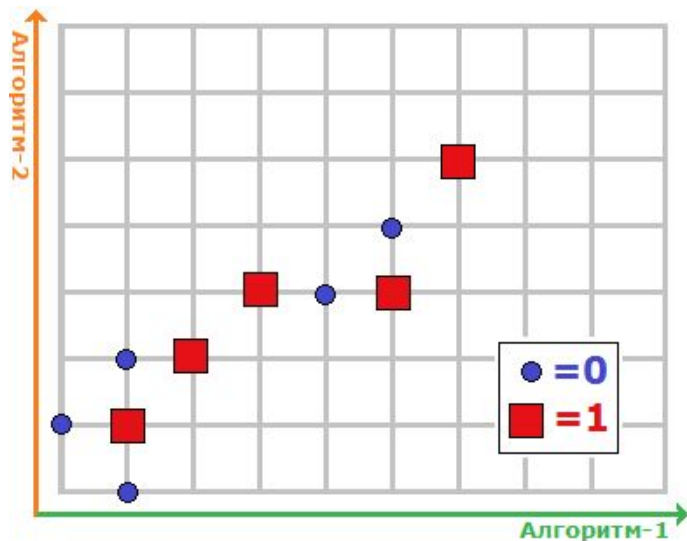
AUC (ROC)

Пример: построить ROC-кривую для предсказаний двух алгоритмов



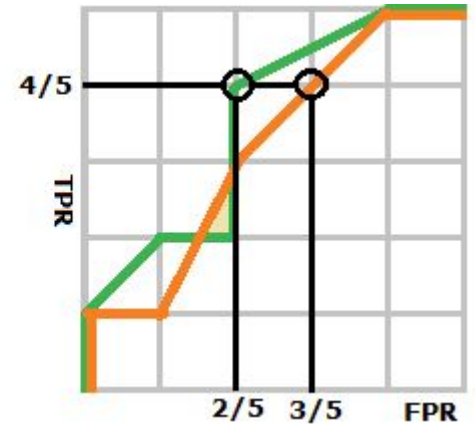
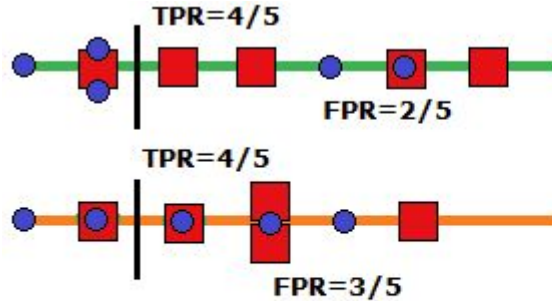
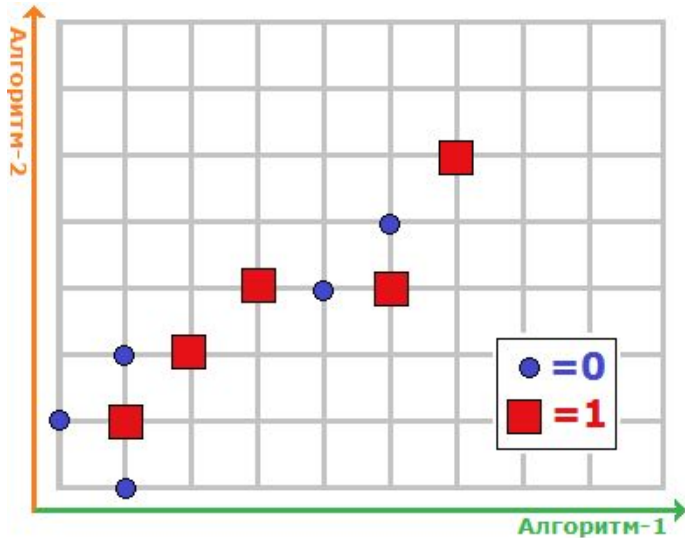
AUC (ROC)

Пример: построить ROC-кривую для предсказаний двух алгоритмов



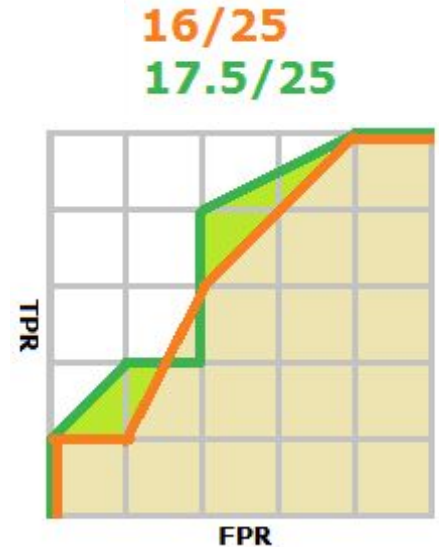
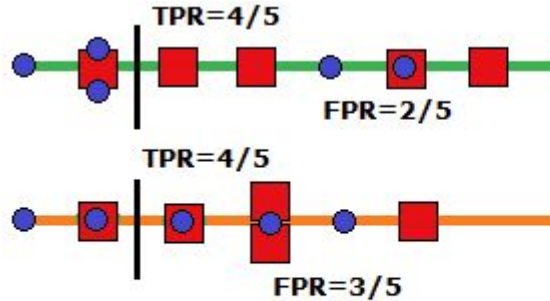
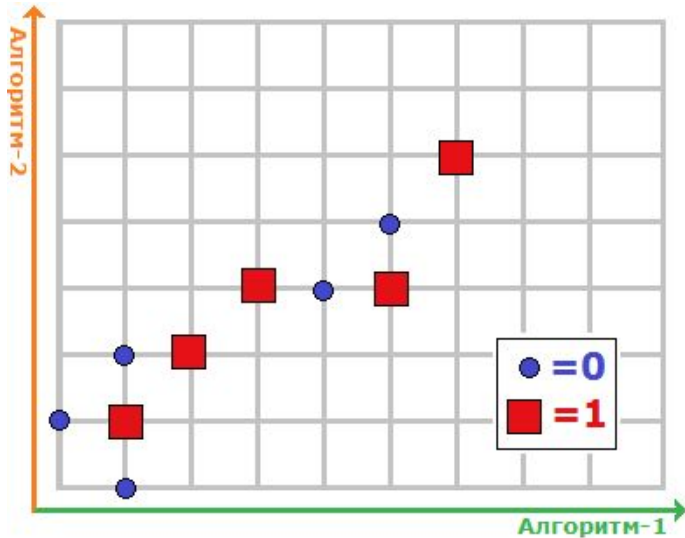
AUC (ROC)

Пример: построить ROC-кривую для предсказаний двух алгоритмов



AUC (ROC)

Пример: построить ROC-кривую для предсказаний двух алгоритмов



Logloss

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

Прогноз - действительное число от 0 до 1

Лучшее константное предсказание - среднее,
то есть частота класса 1

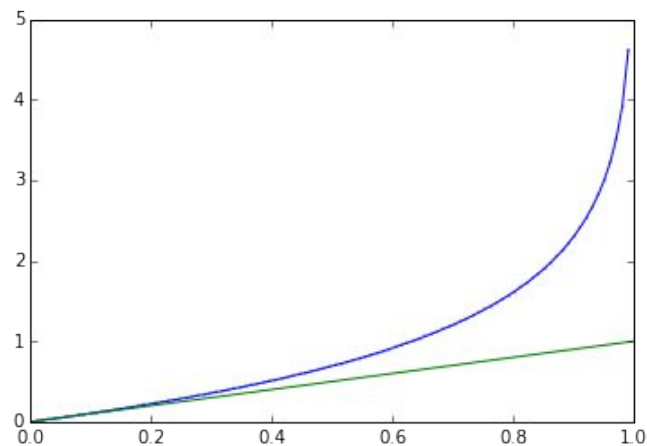
Logloss

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

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



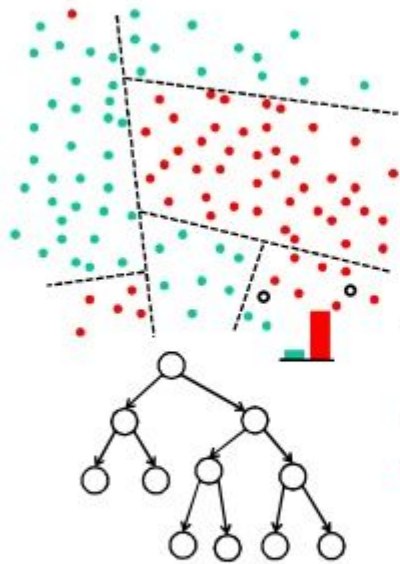
По X: abs(ytrue - ypred)

По Y: LogLoss

Оптимизация метрик

Алгоритмы оптимизируют “свою” метрику

Задача регрессии, `sklearn.tree.DecisionTree`

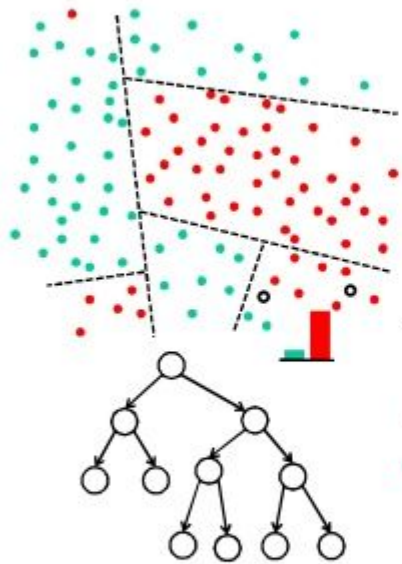


Решаем задачу с метрикой MAE,
но `DecisionTree` оптимизирует MSE.

Что нужно сделать, чтобы оптимизировалось MAE?

Алгоритмы оптимизируют “свою” метрику

Задача регрессии, `sklearn.tree.DecisionTree`



Решаем задачу с метрикой MAE,
но `DecisionTree` оптимизирует MSE.

Что нужно сделать, чтобы оптимизировалось MAE?

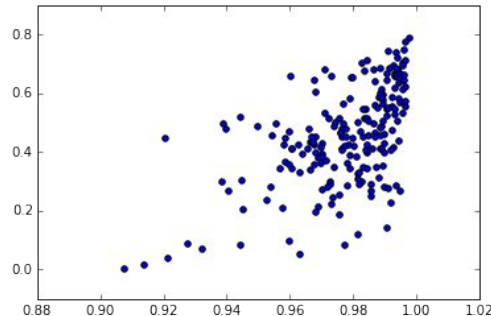
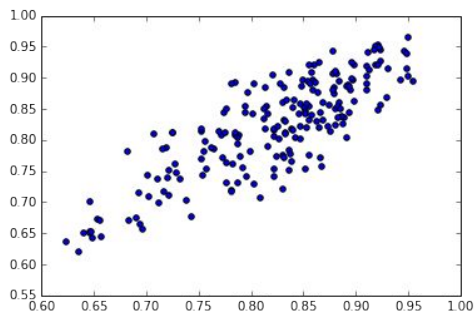
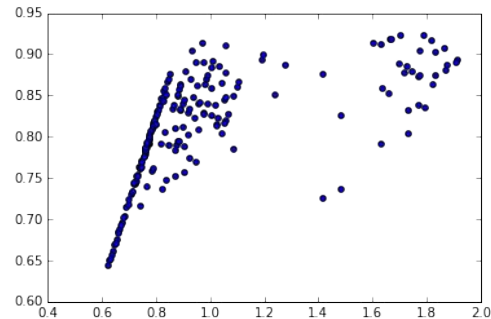
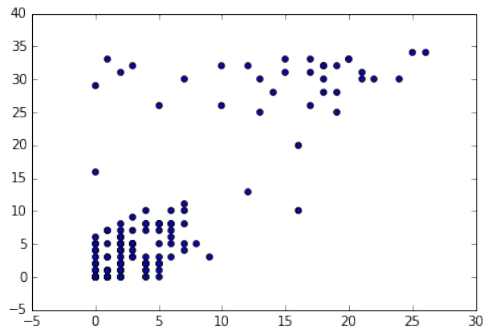
Ответ: Изменить критерий разбиения выборки в
вершинах дерева.

Как смешивать?

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

2. Смешивание рангов

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



Как смешивать?

1. Метрики, чувствительные к значению:

RMSE, Logloss, etc

$y = y_1 * w_1 + y_2 * (1 - w_1)$ # взвешенное среднее

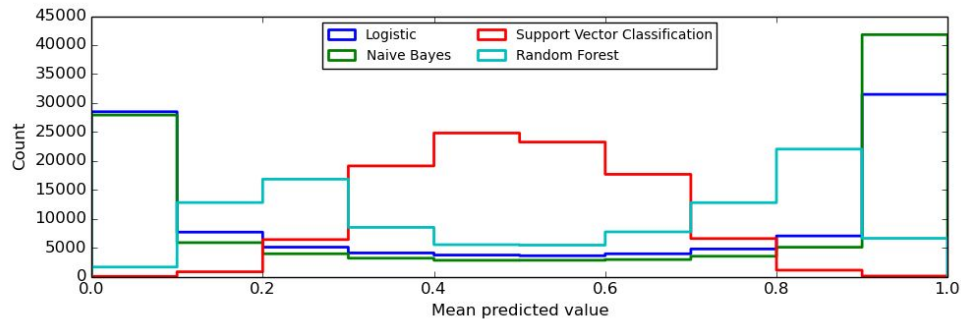
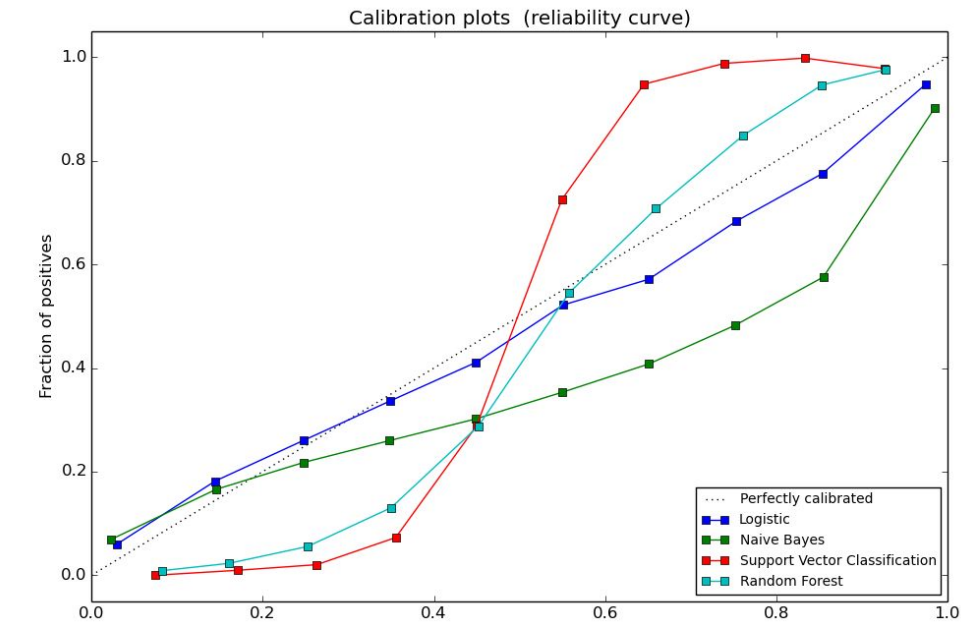
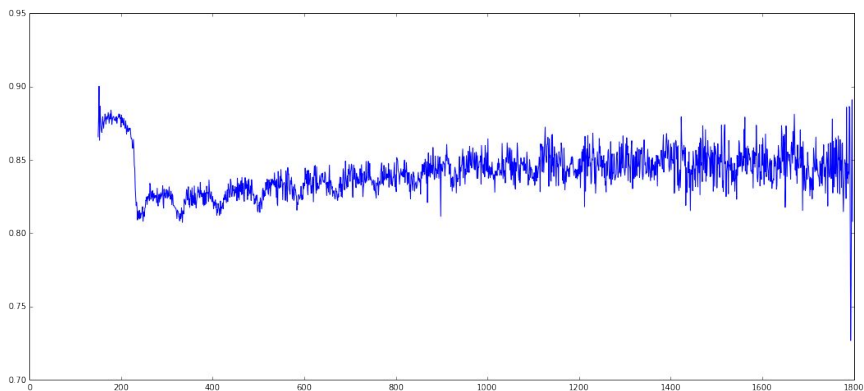
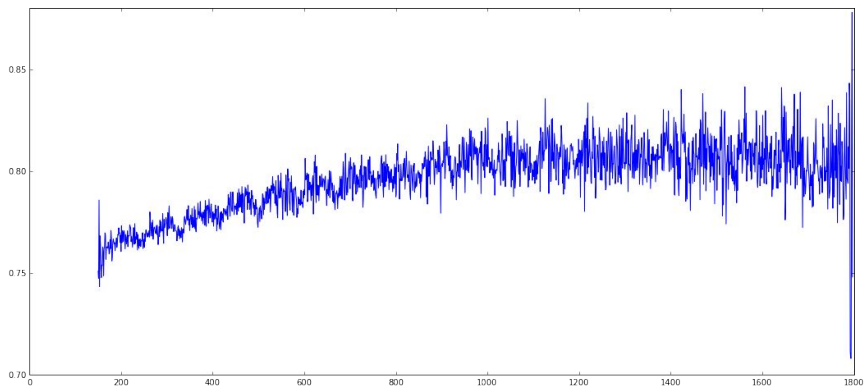
$y = \text{np.sqrt}(y_1 * y_2)$ # геометрическое среднее

2. Метрики, чувствительные к порядку:

AUC (ROC)

$y = y_1^{** 2} + 0.3 * y_2^{** 0.5}$ # всё, что угодно

Probability calibration (Axa)



Разные распределения у в трейне и тесте

`y_train.mean() = 0.007`

`y_test_public.mean() = 0.035` `#публичный тест в соревнованиях`

`y_test_private.mean() = ?` `#приватный тест в соревнованиях`

Как сделать оценку `y.mean()` с 1 сабмита?

Без сабмитов?

`xgboost: scale_pos_weights`

Ссылки

Задачи на ROC AUC:

<https://alexanderdyakonov.wordpress.com/2015/10/09/задачи-про-auc-roc/>

Understanding ROC curves:

<http://www.navan.name/roc/>