



Оценка качества моделей и работа с признаками

Занятие №4

Журавлёв Вадим

@ технопарк

Блоги Люди Программа Выпуски Расписание Вакансии

Вадим Журавлёв

Показывать

Ближайшие две недели Весь семестр

Дисциплина

Основы машинного обучения

Тип события

Все типы

Группа

Все группы

30 сентября 18:00 — 21:00 Основы машинного обучения Смешанное занятие 1 Уточняется среда ML-11

7 октября 18:00 — 21:00 Основы машинного обучения Смешанное занятие 2 Уточняется среда ML-11

14 октября 18:00 — 21:00 Основы машинного обучения Смешанное занятие 3 Уточняется среда ML-11

21 октября 18:00 — 21:00 Основы машинного обучения Смешанное занятие 4 Уточняется среда ML-11

28 октября 18:00 — 21:00 Основы машинного обучения Смешанное занятие 5 Уточняется среда ML-11

3 ноября 18:00 — 21:00 Основы машинного обучения Смешанное занятие 6 Уточняется вторник ML-11

11 ноября 18:00 — 21:00 Основы машинного обучения Смешанное занятие 7 Уточняется среда ML-11

сентябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

октябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2	3	4	
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

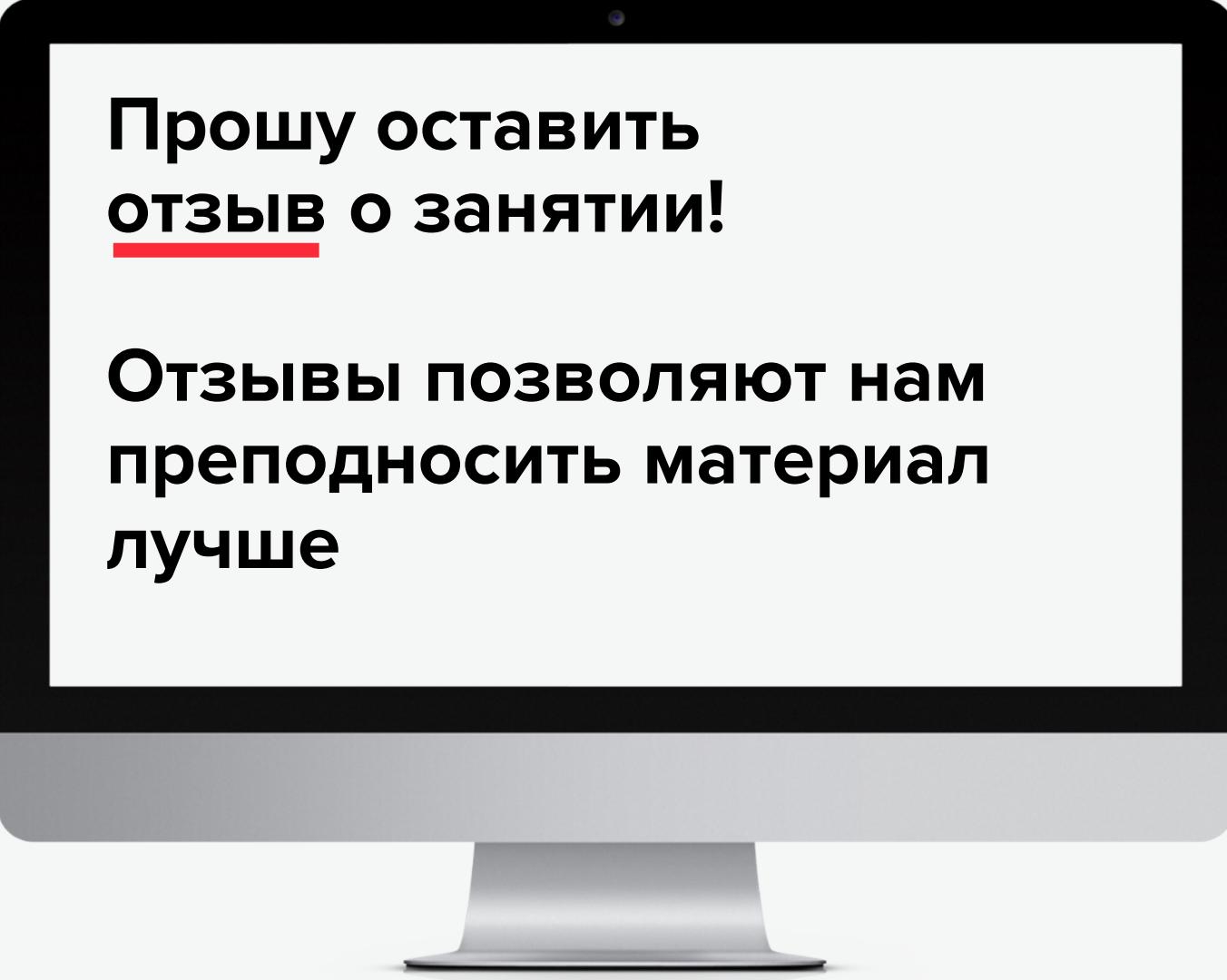
ноябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
						1

Приходя на лекцию



отметиться не
забудь ты



**Прошу оставить
отзыв о занятии!**

**Отзывы позволяют нам
преподносить материал
лучше**

Содержание занятия

1. Метрики качества

1.1. Метрики качества

1.2. Оценка качества моделей

2. Работа с признаками

2.1. Извлечение признаков

2.2. Преобразование признаков

2.3. Работа с пропущенными данными

2.4. Отбор признаков

Метрики



If you can not
measure it,
you can not
improve it.



- Lord Kelvin

Зачем нужны метрики качества?

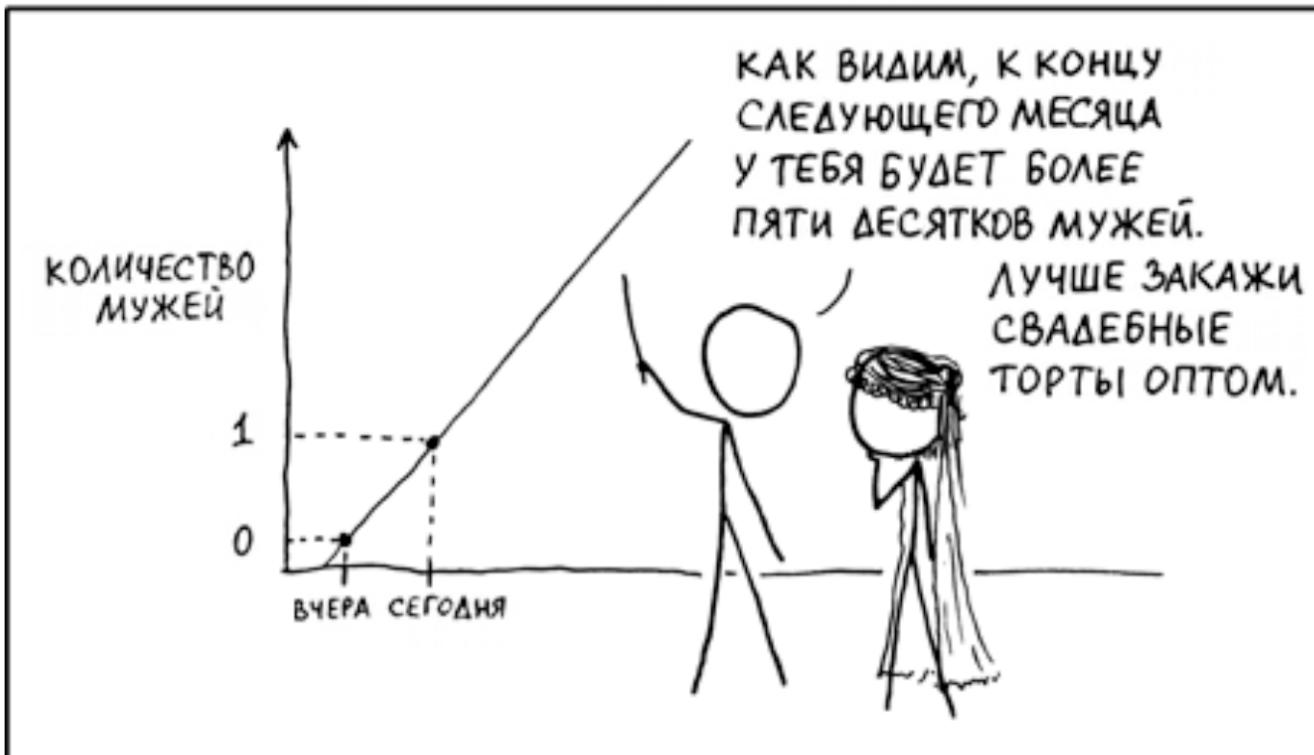
Для оценки качества работы модели

Для сравнения моделей

Для интерпретации результатов

Метрики регрессии

МОЁ ХОББИ: ЭКСТРАПОЛИРОВАТЬ



Постановка задачи

X - множество **объектов**;

$Y \in \mathbb{R}$ - множество **ответов**;

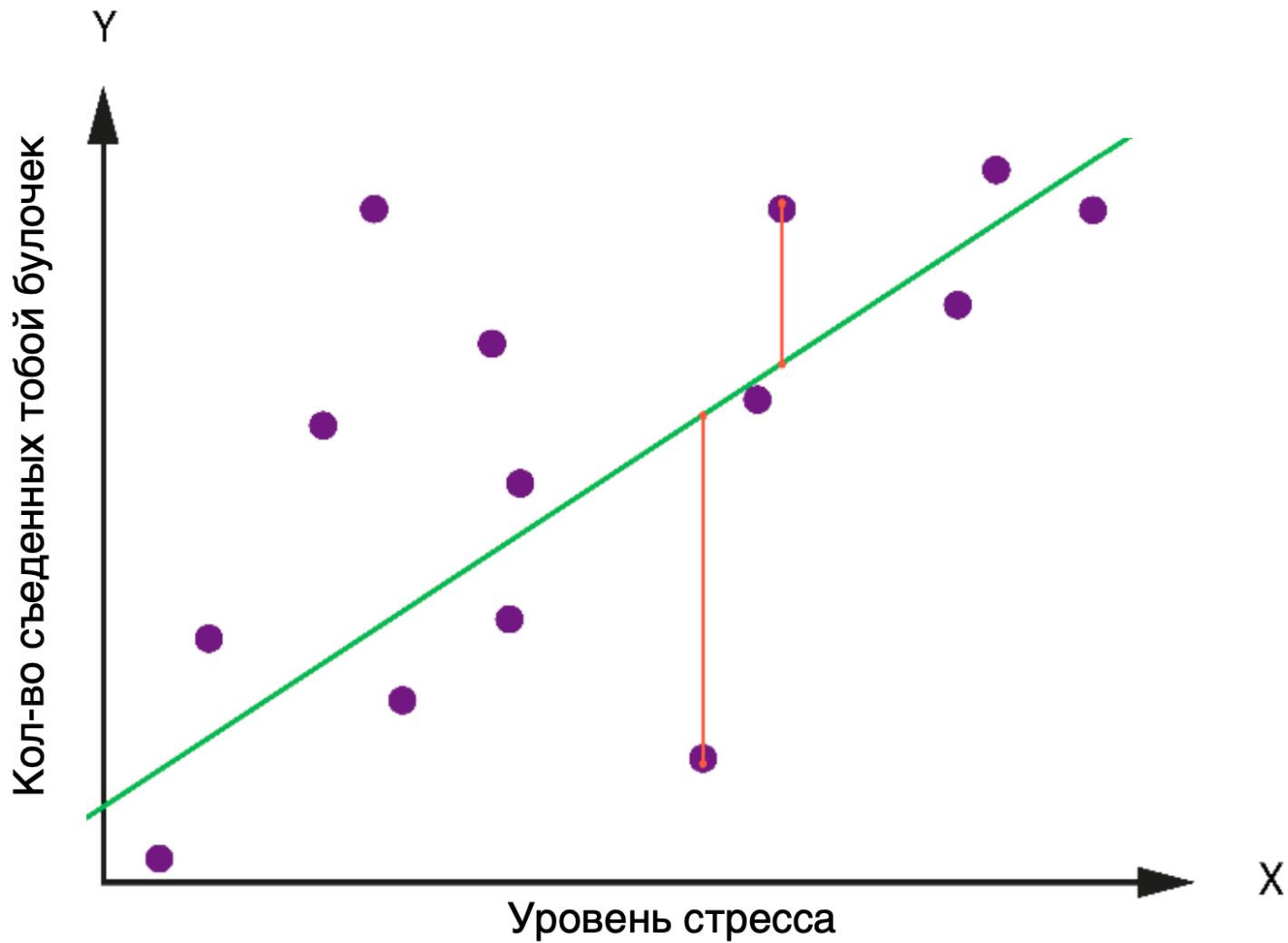
$\{x_1, \dots, x_\ell\} \subset X$ - обучающая **выборка**

$y_i = y(x), i = 1, \dots, \ell$ - известные **ответы**

$a : X \rightarrow Y$ - **алгоритм**, решающий функцию (decision function),
приближающую y на всем мн-же X

$a_i = a(x), i = 1, \dots, \ell$ - **ответы** нашего алгоритма (предсказанное
значение)

Остатки

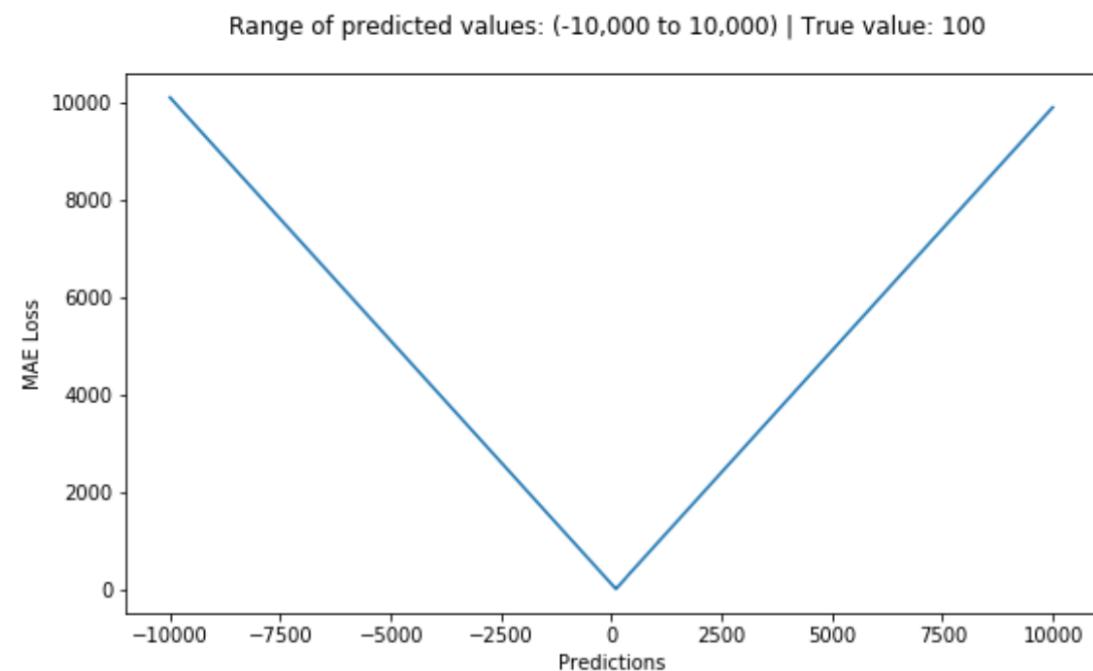


MAE (Mean Absolute Error)

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

C = target median

- Единицы измерения как у таргета
- Сложно интерпретировать
- Нечувствителен к выбросам
- Не дифференцируемая



MAE (Mean Absolute Error)

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

Лучшее константное предсказание - медиана

\hat{Y}	Y
0.1	0
0.5	1
0.6	1
0.5	1
0.3	0

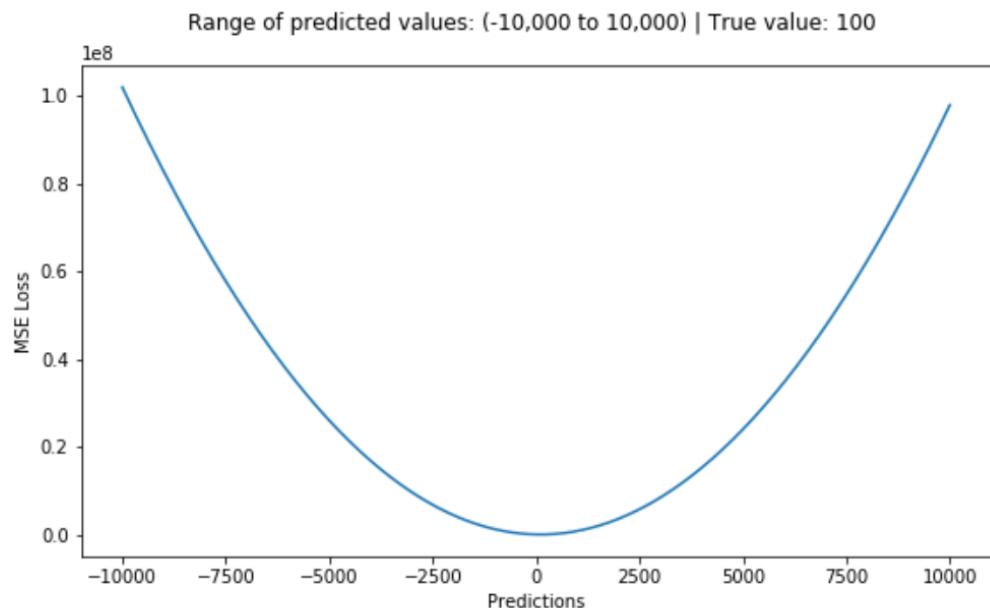
$$MAE = \frac{1}{5}(|0.1 - 0| + |0.5 - 1| + |0.6 - 1| + |0.5 - 1| + |0.3 - 0|)$$

MSE (Mean Squared Error)

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

C = target mean

- Дифференцируемая
- Чувствительна к выбросам
- Сложно интерпретировать



RMSE (Root Mean Squared Error)

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

C = target mean

- Дифференцируемая
- Чувствительна к выбросам
- Интерпретация: стандартное отклонение ответа

MSPE MAPE

Mean Squared Percent Error

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

C = weighted target mean

Mean Absolute Percent Error

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

C = weighted target median

- Присваивают больший вес абсолютно более маленьким объектам => смещены.
- Нечувствительны к выбросам.
- Хорошо интерпретируются: относительный прирост.

RMSLE (Root Mean Squared Logarithmic Error)

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

C = exp(target mean)

- RMSLE = RMSE in log space
- RMSLE иногда противопоставляют MAPE, так как она менее смещена по отношению маленьким объектам

R^2 (коэффициент детерминации)

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

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

- Дифференцируемый
- Чувствителен к выбросам
- Хорошо интерпретируется: насколько наша модель лучше, чем константное решение

Выводы

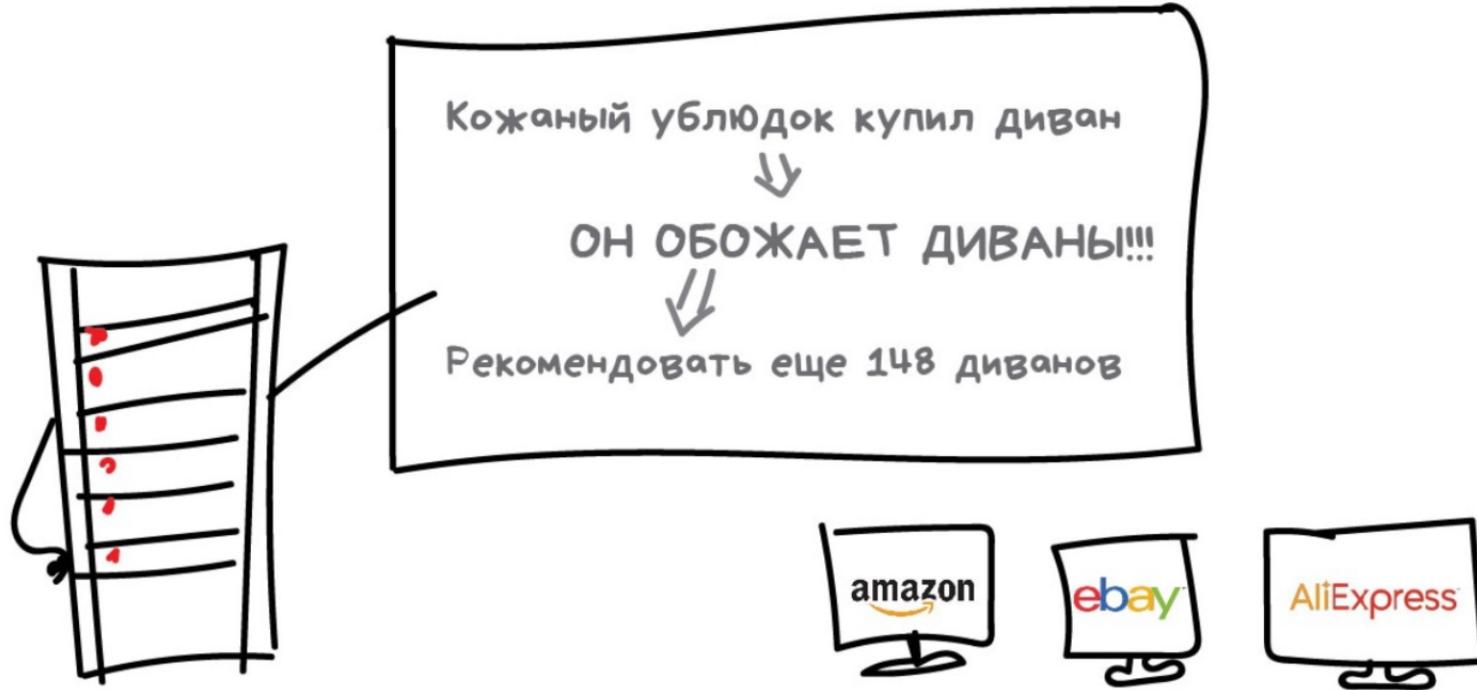
Аномальные значения - лишь неожиданные значения, которые нужно учитывать?

Используем MSE, RMSE или R^2 для интерпретации результатов.

Аномальные значения - это выбросы?

Вычищаем их или используем MAE или MAPE для интерпретации результатов.

Метрики классификации



Постановка задачи

X - множество объектов;

$Y \in \mathbb{R}$ - множество ответов;

$\{x_1, \dots, x_\ell\} \subset X$ - обучающая выборка

$y_i = y(x), i = 1, \dots, \ell$ - известные ответы

$a : X \rightarrow Y$ - алгоритм, решающий функцию (decision function),
приближающую y на всем множестве X

$a_i = a(x), i = 1, \dots, \ell$ - ответы нашего алгоритма (метка класса или
вероятность)

ACCURACY (доля правильных ответов)

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N [\hat{y}_i = y_i]$$

Подсчитываем долю правильно предсказанных объектов
Может быть использована в многоклассовой классификации

```
target = np.array([1, 1, 1, 2, 1, 1, 1, 2])
pred = np.array([1, 1, 1, 2, 1, 1, 1, 1])
```

Чему равна accuracy?

Accuracy (доля правильных ответов)

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N [\hat{y}_i = y_i]$$

C = самый популярный класс



KFC = 10

Dog = 90

—
Accuracy = ???

Accuracy (доля правильных ответов)

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N [\hat{y}_i = y_i]$$

C = самый популярный класс



KFC = 10

Dog = 90

—
Accuracy = 0.9!

Чувствителен к дисбалансу классов!

Confusion Matrix

		Actual Values	
		1	0
Predicted Values	1	 TRUE POSITIVE	 FALSE POSITIVE
	0	 FALSE NEGATIVE	 TRUE NEGATIVE

H0: человек не ждет ребенка
Ha: ох как ждет :)

Precision и Recall

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Точность - доля беременных среди всех предсказанных моделью беременных

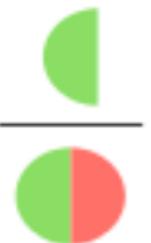
$$\text{precision} = \frac{TP}{TP + FP}$$

Полнота - доля предсказанных моделью беременных среди всех беременных

$$\text{recall} = \frac{TP}{TP + FN}$$

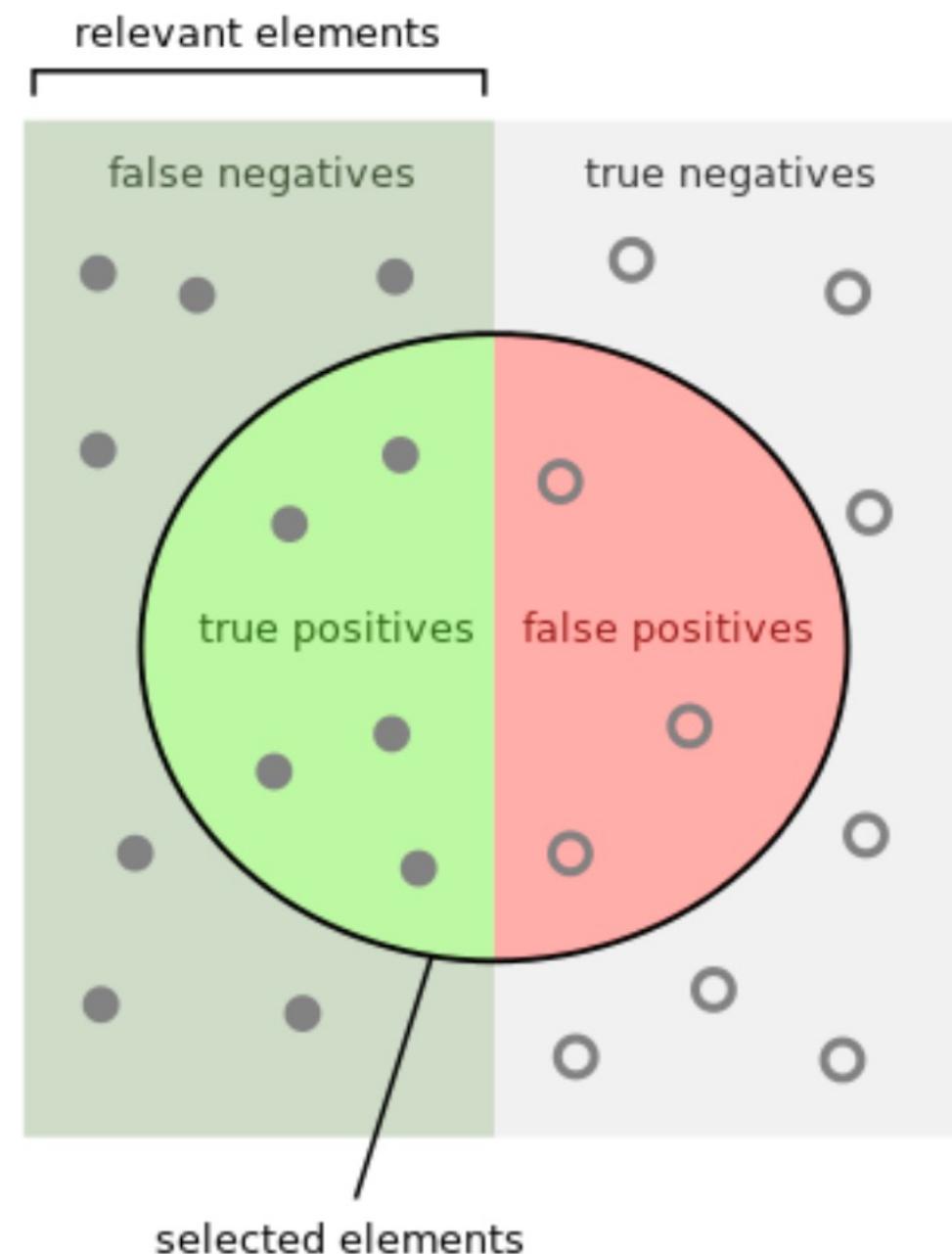
Precision и Recall

How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$


How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Precision и Recall

Recall: Какую часть из объектов класса 1 мы нашли?

Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

Precision и Recall

Recall: Какую часть из объектов класса 1 мы нашли?

Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

1. `target = np.array([0, 1, 1, 0, 1, 1])`
2. `pred = np.array([1, 0, 1, 0, 1, 0])`

Precision и Recall

Recall: Какую часть из объектов класса 1 мы нашли?

Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

1. target = np.array([0, 1, 1, 0, 1, 1])
2. pred = np.array([1, 0, 1, 0, 1, 0])

Результат:

0.5

$2/(2+2)$

0.67

$2/(2+1)$

Recall: Какую часть из объектов класса 1 мы нашли?

Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

Precision и Recall

Recall: Какую часть из объектов класса 1 мы нашли?

Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

1. target = np.array([0, 1, 1, 0, 1, 1])
2. pred = np.array([1, 1, 1, 1, 1, 1])

Precision и Recall

Recall: Какую часть из объектов класса 1 мы нашли?

Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

1. `target = np.array([0, 1, 1, 0, 1, 1])`
2. `pred = np.array([1, 1, 1, 1, 1, 1])`

Результат:

1.0

$4/(4+0)$

0.67

$4/(4+2)$

Recall: Какую часть из объектов класса 1 мы нашли?

Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?





spam

not spam

sent to spam folder

true
positives



sent to inbox

false
negatives



false
positives



true
negatives



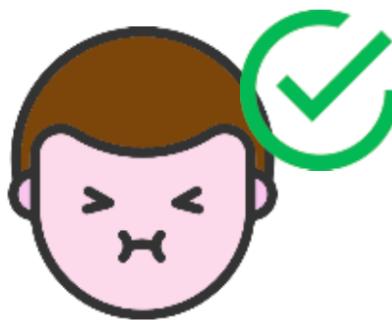


sick

healthy

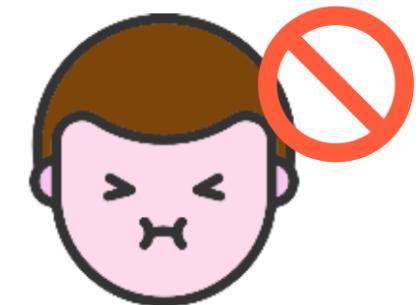
diagnosed sick

true
positives



diagnosed healthy

false
negatives



false
positives



true
negatives





spam

not spam

sent to spam folder

sent to inbox

false
negatives



false
positives





sick

healthy

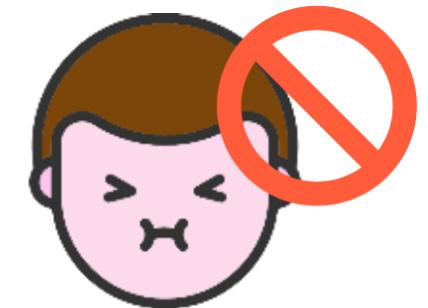
diagnosed sick

diagnosed healthy

false
positives



false
negatives

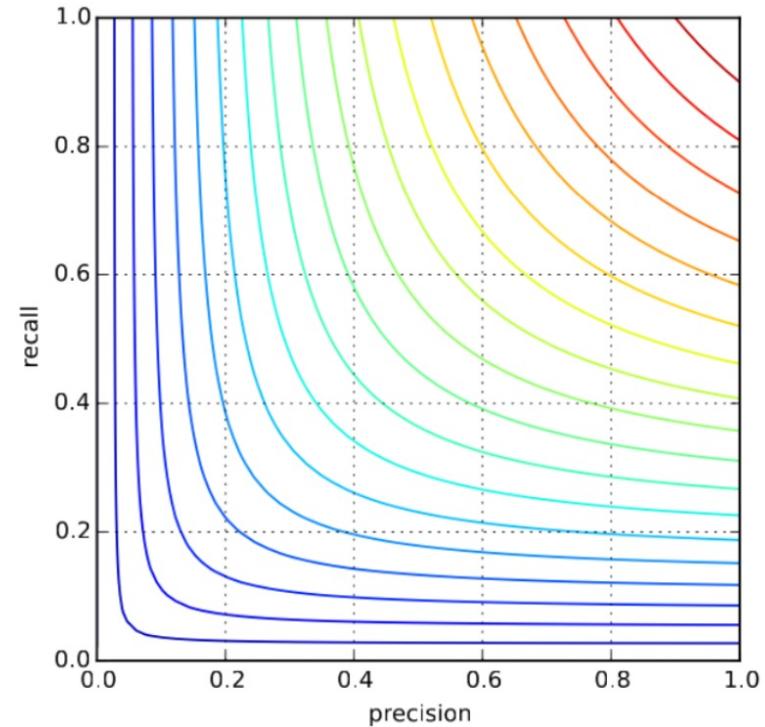


F-measure

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

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{true positive}}{(1 + \beta^2) \cdot \text{true positive} + \beta^2 \cdot \text{false negative} + \text{false positive}}$$

$$F_1 = \left(\frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



Стремится к нулю, когда хотя бы один из аргументов близок к нулю.
 β - определяет важность recall по сравнению с precision.

F-measure

$$F_1 = \left(\frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

1. target = np.array([0, 1, 0, 0, 0, 0])
2. pred = np.array([1, 1, 1, 0, 1, 0])

1. target = np.array([0, 1, 0, 0, 0, 0])
2. pred = np.array([1, 1, 1, 1, 1, 1])

F-measure

$$F_1 = \left(\frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

1. target = np.array([0, 1, 0, 0, 0, 0])
2. pred = np.array([1, 1, 1, 0, 1, 0]) 0.4

1. target = np.array([0, 1, 0, 0, 0, 0])
2. pred = np.array([1, 1, 1, 1, 1, 1]) 0.29

B jupyter notebook



Classification report B sklearn

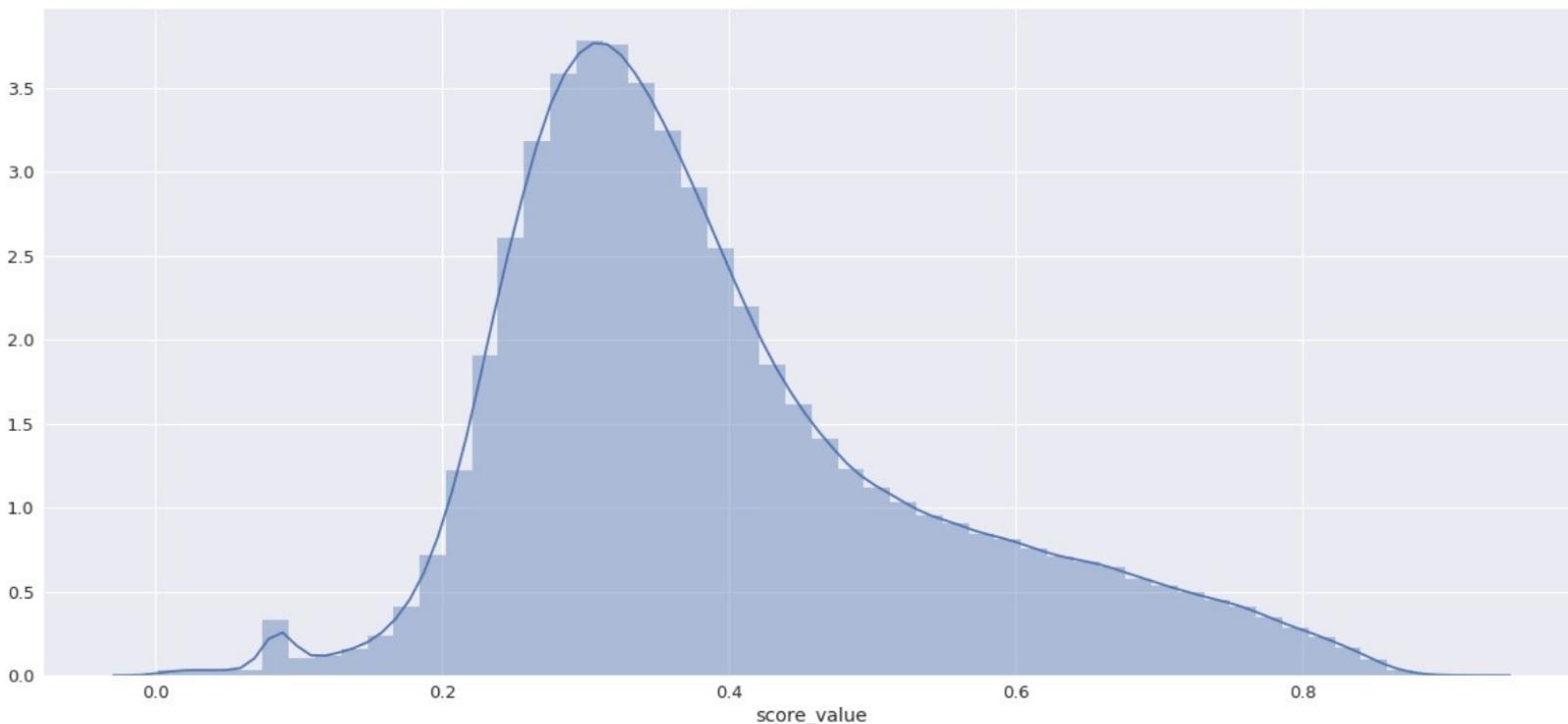
```
from sklearn.metrics import classification_report  
  
print(classification_report(y_test,y_hat_test))
```

		precision	recall	f1-score	support
	0	0.74	0.84	0.79	12733
	1	0.96	0.92	0.94	48532
micro avg		0.91	0.91	0.91	61265
macro avg		0.85	0.88	0.86	61265
weighted avg		0.91	0.91	0.91	61265

Soft target

$a : X \rightarrow Y', Y' \in (0, 1)$ - алгоритм предсказывает значение от 0 до 1
(например, вероятность принадлежности к положительному классу)

Распределение предсказания на тестовом множестве



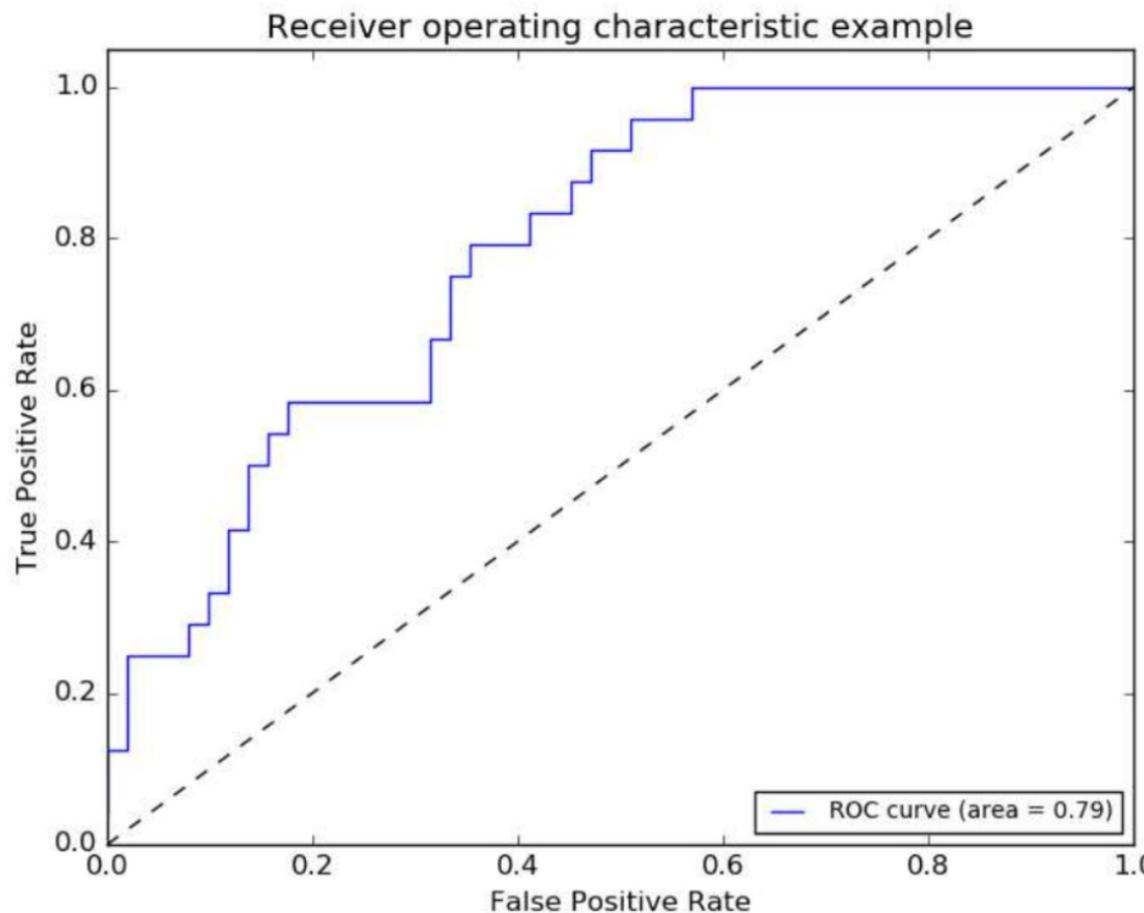
Label	Per-Class F1 Score	Macro-Averaged F1 Score
 Airplane	0.67	$\frac{0.67 + 0.40 + 0.67}{3} = \mathbf{0.58}$
 Boat	0.40	
 Car	0.67	

Label	Per-Class F1 Score	Support	Support Proportion	Weighted Average F1 Score
 Airplane	0.67	3	0.3	$(0.67 * 0.3) + (0.40 * 0.1) + (0.67 * 0.6)$ $= \mathbf{0.64}$
 Boat	0.40	1	0.1	
 Car	0.67	6	0.6	
Total	-	10	1.0	

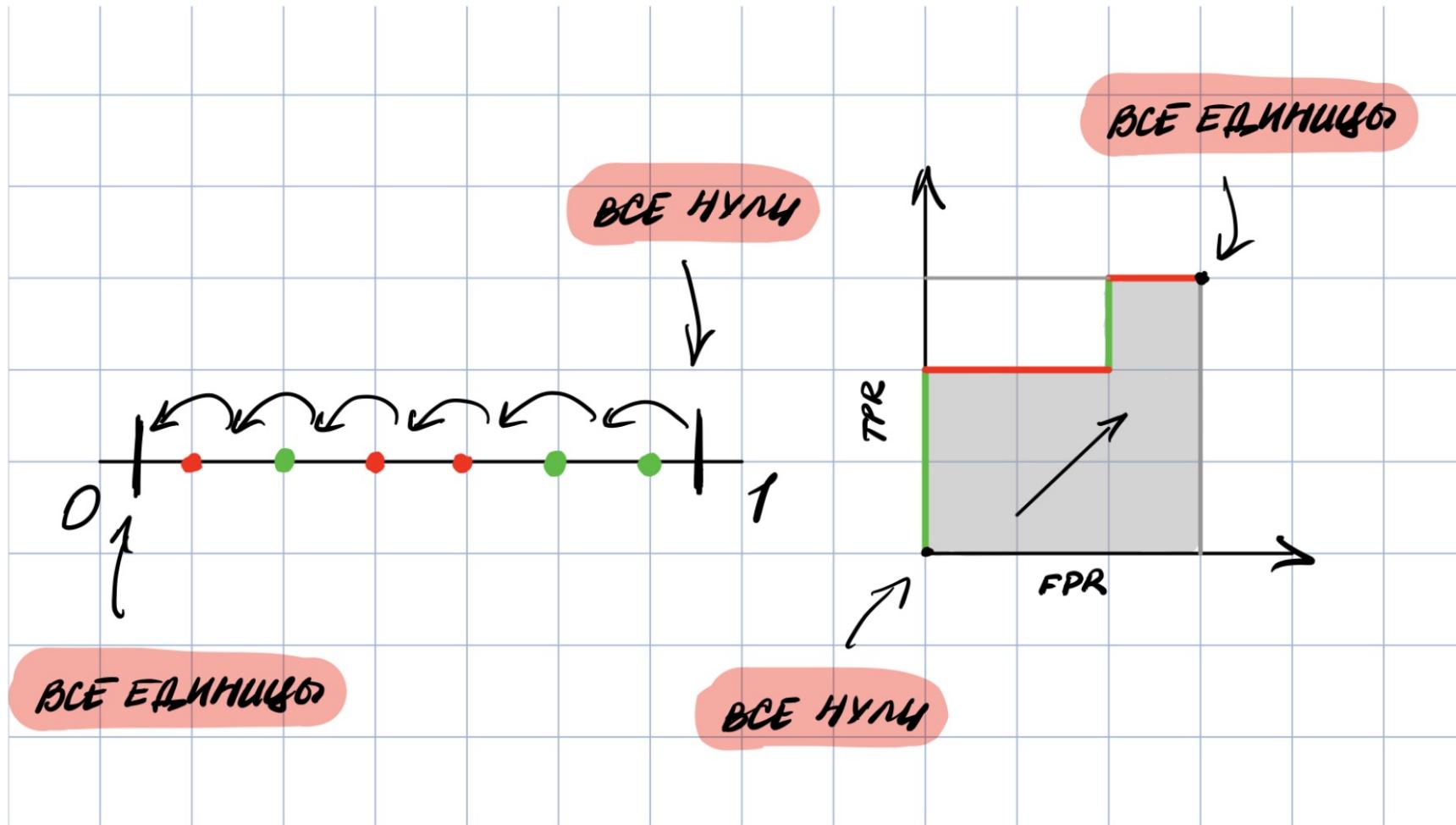
Label	True Positive (TP)	False Positive (FP)	False Negative (FN)	Micro-Averaged Values
 Airplane	2	1	1	$\text{Precision} = \frac{6}{6+4} = \mathbf{0.60}$ $\text{Recall} = \frac{6}{6+4} = \mathbf{0.60}$ $\text{F1 Score} = \frac{6}{6 + \frac{1}{2}(4+4)} = \mathbf{0.60}$
 Boat	1	3	0	
 Car	3	0	3	
TOTAL	6	4	4	

ROC AUC (ROC Area Under Curve)

Определяет долю правильно отранжированных пар



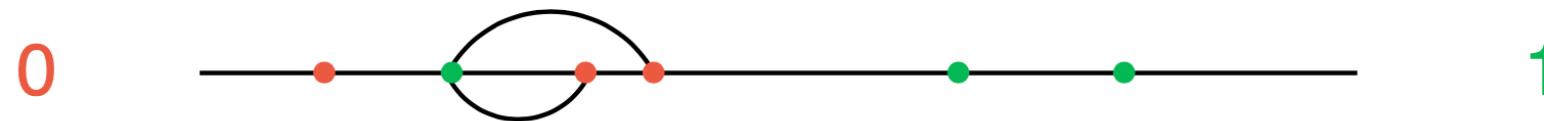
ROC AUC



$$\text{ROC-AUC} = 7 / 9$$

*только пары с разными метками

$$\text{AUC} = \frac{\# \text{ correctly ordered pairs}}{\text{total number of pairs}} = 1 - \frac{\# \text{ incorrectly ordered pairs}}{\text{total number of pairs}}$$
$$= 1 - 2 / 9 = 7 / 9$$



Можно сделать вывод, что метрика roc auc нечувствительна на выборках с несбалансированными классами.

Чуть более подробно:

id	оценка	класс
1	0.5	0
2	0.1	0
3	0.2	0
4	0.6	1
5	0.2	1
6	0.3	1
7	0.0	0

Табл. 1

Чуть более подробно:

id	оценка	класс
1	0.5	0
2	0.1	0
3	0.2	0
4	0.6	1
5	0.2	1
6	0.3	1
7	0.0	0

Табл. 1

id	оценка	класс
4	0.6	1
1	0.5	0
6	0.3	1
3	0.2	0
5	0.2	1
2	0.1	0
7	0.0	0

Табл. 2

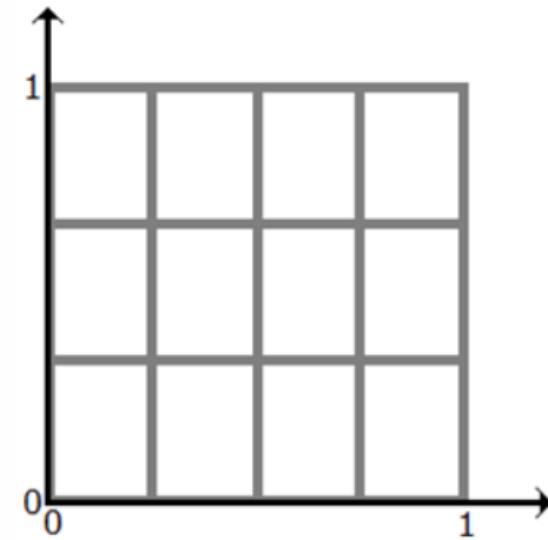
Чуть более подробно:

id	оценка	класс
1	0.5	0
2	0.1	0
3	0.2	0
4	0.6	1
5	0.2	1
6	0.3	1
7	0.0	0

Табл. 1

id	оценка	класс
4	0.6	1
1	0.5	0
6	0.3	1
3	0.2	0
5	0.2	1
2	0.1	0
7	0.0	0

Табл. 2



Важный момент: если у нескольких объектов значения оценок равны, то мы делаем шаг в точку, которая на a блоков выше и b блоков правее, где a – число единиц в группе объектов с одним значением метки, b – число нулей в ней. В частности, если все объекты имеют одинаковую метку, то мы сразу шагаем из точки $(0, 0)$ в точку $(1, 1)$.

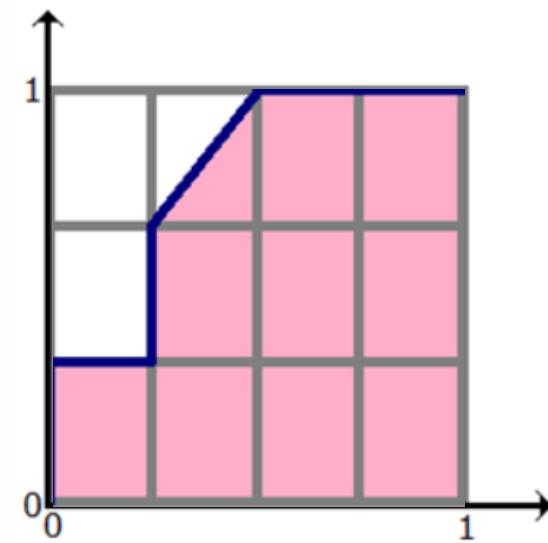
Чуть более подробно:

id	оценка	класс
1	0.5	0
2	0.1	0
3	0.2	0
4	0.6	1
5	0.2	1
6	0.3	1
7	0.0	0

Табл. 1

id	оценка	класс
4	0.6	1
1	0.5	0
6	0.3	1
3	0.2	0
5	0.2	1
2	0.1	0
7	0.0	0

Табл. 2



ROC-AUC

True = [0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1]

Pred = [0.2, 0.7, 0.6, 0.1, 0.5, 0.3, 0.5, 0.4, 0.6, 0.1, 0.1, 0.1, 0.9, 0.8, 0.1, 0.5, 0.3, 0.2]

ROC-AUC

True = [1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0]

Pred = [0.9, 0.8, 0.7, 0.6, 0.6, 0.5, 0.5, 0.5, 0.5, 0.4, 0.3, 0.3, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1]

ROC-AUC

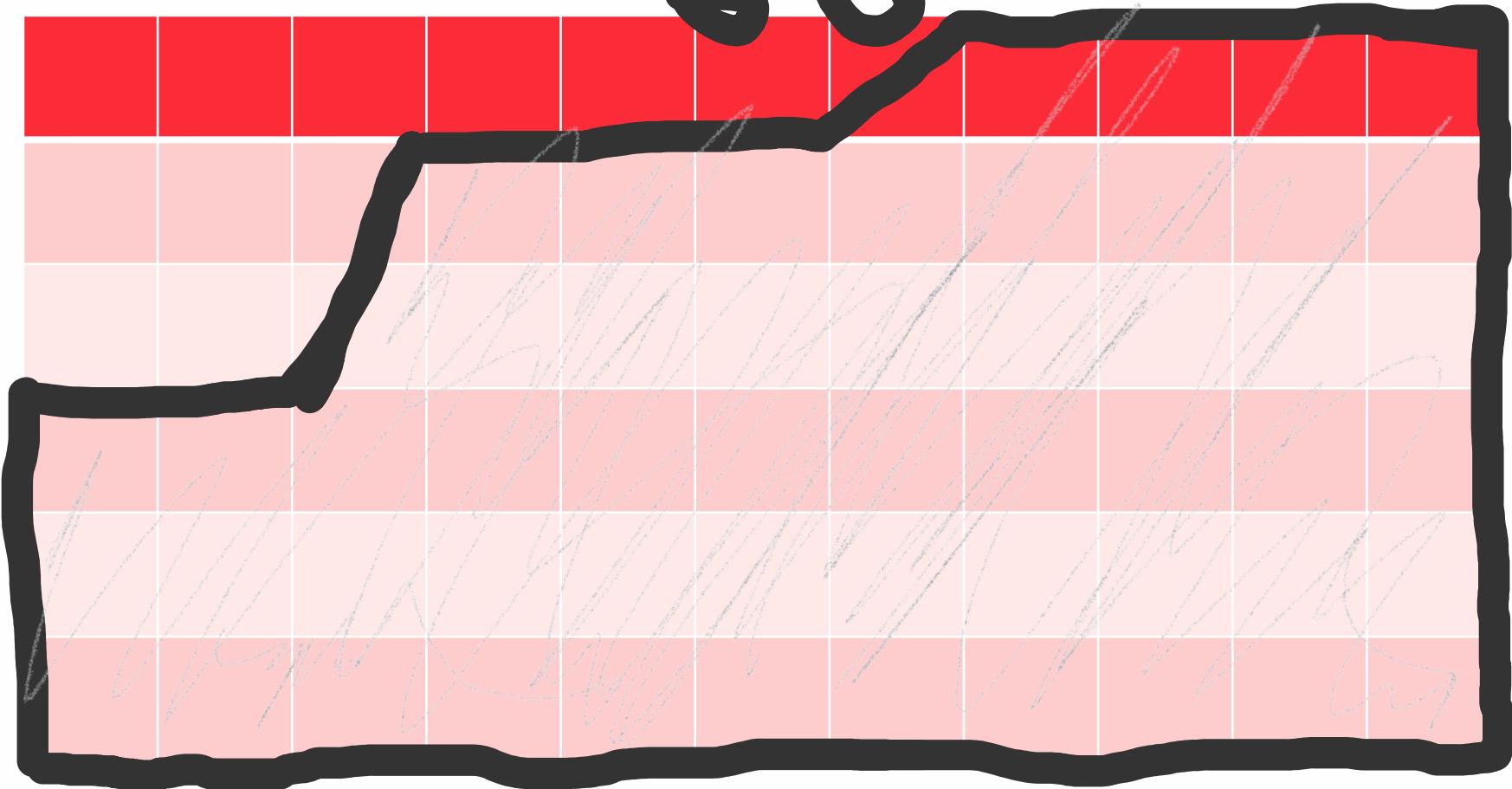
True = [1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0]

Pred = [0.9, 0.8, 0.7, 0.6, 0.6, 0.5, 0.5, 0.5,
0.4, 0.3, 0.3, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1]

PRED	TRUE
0.9	1
0.8	1
0.7	1
0.6	0
0.6	0
0.5	1
0.5	0
0.5	1
0.4	0
0.3	0
0.3	0
0.2	0
0.2	1
0.1	0
0.1	0
0.1	0
0.1	0

ROC-AUC

54.
5
—
66



PRED	TRUE
0.9	1
0.8	1
0.7	1
0.6	0
0.6	0
0.5	1
0.5	0
0.5	1
0.4	0
0.3	0
0.3	0
0.2	0
0.2	1
0.1	0
0.1	0
0.1	0
0.1	0

Смысл ROC-AUC

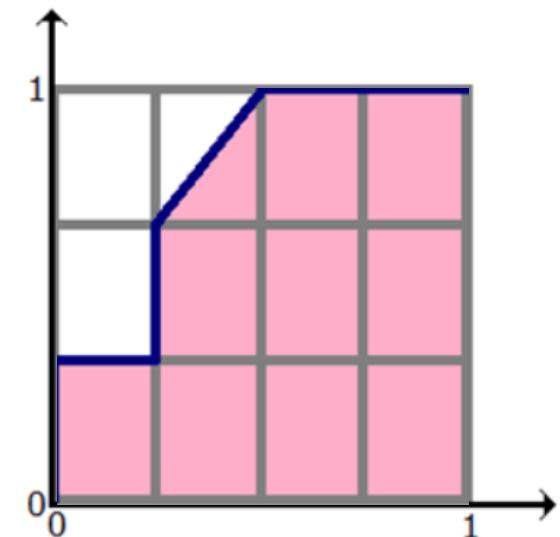
Сетка на разбила квадрат на $m \times n$ блоков. Ровно столько же пар вида (объект класса 1, объект класса 0), составленных из объектов тестовой выборки. Каждый закрашенный блок на рисунке соответствует паре (объект класса 1, объект класса 0), для которой наш алгоритм правильно предсказал порядок (объект класса 1 получил оценку выше, чем объект класса 0), незакрашенный блок – паре, на которой ошибся.

Таким образом, AUC ROC равен доле пар объектов вида (объект класса 1, объект класса 0), которые алгоритм верно упорядочил, т.е. первый объект идёт в упорядоченном списке раньше. Численно это можно записать так:

$$\frac{\sum_{i=1}^q \sum_{j=1}^q I[y_i < y_j] I[a_i < a_j]}{\sum_{i=1}^q \sum_{j=1}^q I[y_i < y_j]}, \quad (*)$$

$$I'[a_i < a_j] = \begin{cases} 0, & a_i > a_j, \\ 0.5 & a_i = a_j, \\ 1, & a_i < a_j, \end{cases} \quad I[y_i < y_j] = \begin{cases} 0, & y_i \geq y_j, \\ 1, & y_i < y_j, \end{cases}$$

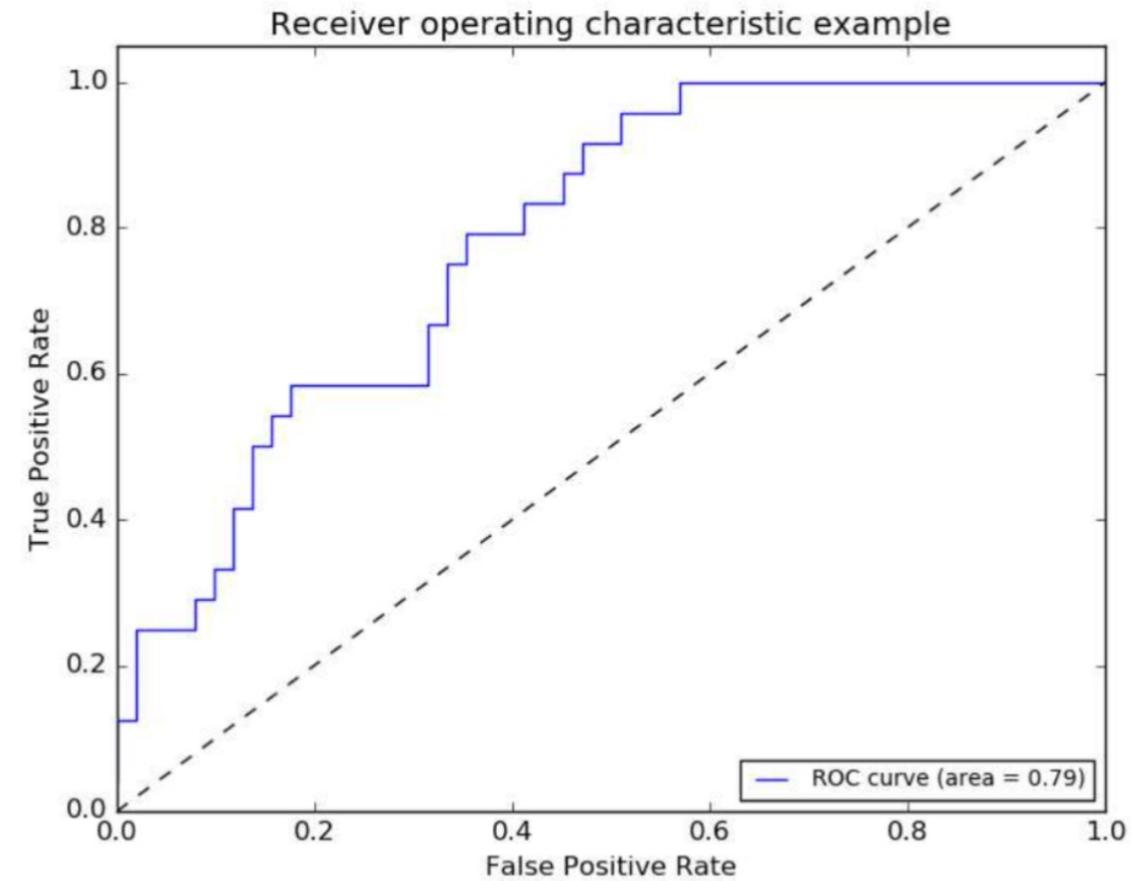
a_i – ответ алгоритма на i -м объекте, y_i – его метка (класс), q – число объектов в teste.



Индекс Джини

$$\text{GINI} = 2 * \text{ROC-AUC} - 1$$

По сути это площадь между
ROC-кривой и диагональю
соединяющей точки $(0,0)$ и $(1, 1)$



LogLoss

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

- Дифференцируем!
- Позволяет корректно предсказывать вероятности
- Есть почти во всех методах

Logloss

Неприятное свойство:

если для объекта 1-го класса мы предсказываем нулевую вероятность принадлежности к этому классу или, наоборот, для объекта 0-го – единичную вероятность принадлежности к классу 1, то ошибка равна **бесконечности!** Таким образом, **грубая ошибка на одном объекте сразу делает алгоритм бесполезным.** На практике часто логлосс ограничивают каким-то большим числом (чтобы не связываться с бесконечностями).

Оценка качества моделей

Как выбрать лучшую модель для решения конкретной задачи?



Проблема переобучения

Способы обучиться на наборе данных:

1. Найти общие закономерности в предоставленном наборе данных
2. Запомнить правильные ответы



Для оценки качества модели нельзя использовать те же данные, что и для построения модели.

Train, Test, Validation

- На **Train** обучаем модели-кандидаты
-
- На **Test** оцениваем модели-кандидаты и выбираем
 - лучшую
- На **Validation** проверяем, что все работает как ожидалось
- **Validation** никак не используем при построении модели!
- На гиперпараметрах модели тоже можно переобучиться!

Shuffle & Split

Перемешиваем samples, делим датасет на две части (**Train** и **Test**) в некоторой пропорции.

На **Train** обучаем модель, на **Test** оцениваем качество.

Особенности:

- Простая реализация
- Разумно использовать когда данных "много"

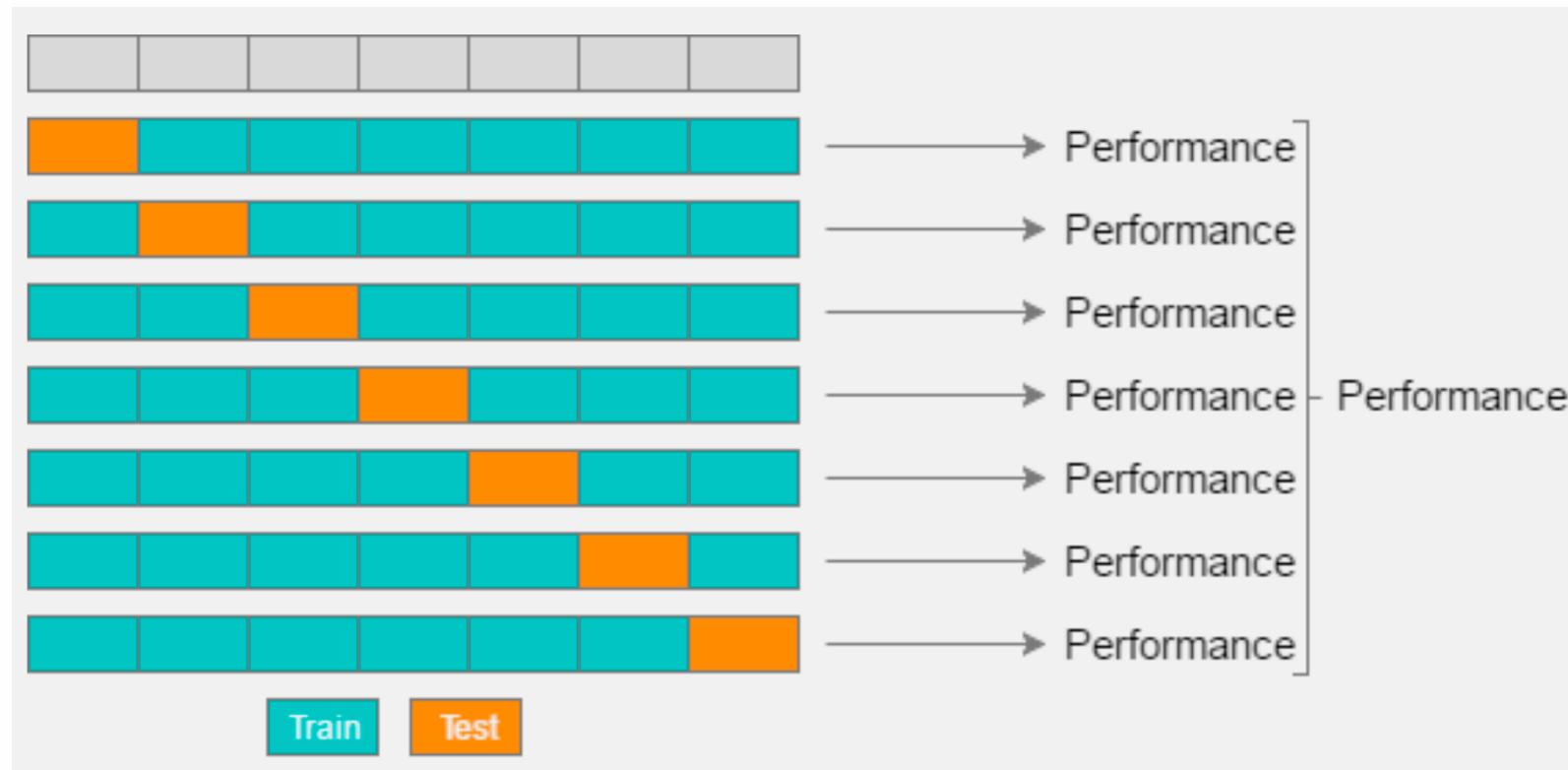
K-fold

Разбиваем датасет на **K** равных частей, затем строим **K** моделей где в качестве **Test** берем одну из частей, а все остальные используем как **Train**. На **Train** обучаем модель, на **Test** оцениваем качество.

Особенности:

- Используем все данные как для построения моделей, так и для оценки качества
- Один из наиболее популярных методов оценки качества моделей

K-fold



Leave One Out

Экстремальный случай **K-fold**, когда **K** равно числу сэмплов в наборе данных.

Особенности:

- Модель на датасете без одного сэмпла практически идентична модели на полном датасете
- Может быть эффективно посчитан для некоторых видов моделей

Повторные разбиения

K-fold или **Shuffle & Split**, повторенный **N** раз с различными разбиениями.

Особенности:

- В **N** раз выше вычислительная сложность
- Эффективны когда данных "мало" или данные "шумные"

Стратифицированные разбиения

Стратифицированные разбиения - это такие разбиения, которые сохраняют определенные свойства исходной выборки.

Свойствами могут быть:

- Распределение целевой переменной
- Распределения некоторых признаков

Стратифицированные разбиения

Плюсы:

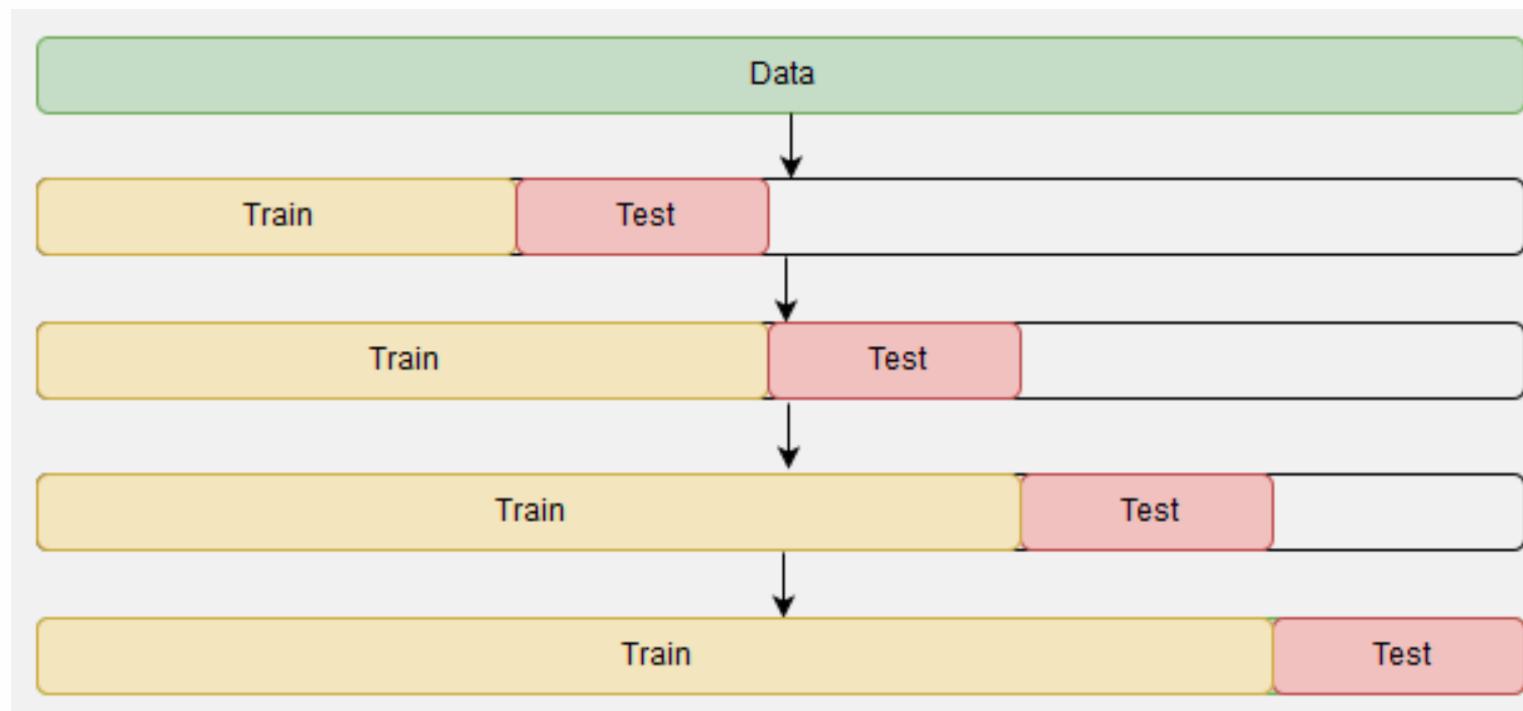
- Могут обеспечить большую точность, чем простой случайный выбор
- Позволяют избежать "непредставительной" выборки (например отсутствие какого-либо класса объектов в разбиении)

Минусы:

- Сложнее в реализации

Разбиение временных рядов

Разбиение временных рядов следует проводить по времени события.



Работа с признаками



Извлечение признаков

Описываем реальный мир на языке,
понятном модели.



Извлечение признаков

- **Сампл (пример)** - это вектор чисел.
- **Объект** - представлен набором данных.
- **Извлечение признаков** - представление реального или цифрового объекта в виде вектора чисел.

Типы признаков

1) Бинарные (флаг подключения услуги)

Binary $\{true, false\}$

2) Номинальные (тарифный план)

Categorical множество значений конечно

3) Качественные (количество мегабайт в месяц) Numerical

\mathbb{R}

4) Порядковые (месяцы, этажи)

Ordinal множество значений конечно и упорядочено

Извлечение признаков

Задача: Необходимо спрогнозировать расходы абонента при переходе на новый ТП

Признаки, характеризующие расходы клиента на связь:

Бинарные	Номинальные	Количественные	Порядковые
Наличие интернета	Тарифный план Город Тип устройства ОС	Количество звонков Количество сообщений Объем траффика	LTV (время жизни клиента)

Данные

Данные бывают:

1. Числовые
2. Дата и время
3. Геоданные (latitude, longitude)
4. Временные ряды
5. Текстовые данные
6. Графические изображения
7. Звук
8. Видео
9. И др.

Геоданные

Что можно извлечь:

1. Что находится по заданной координате
2. Расстояния до особых объектов

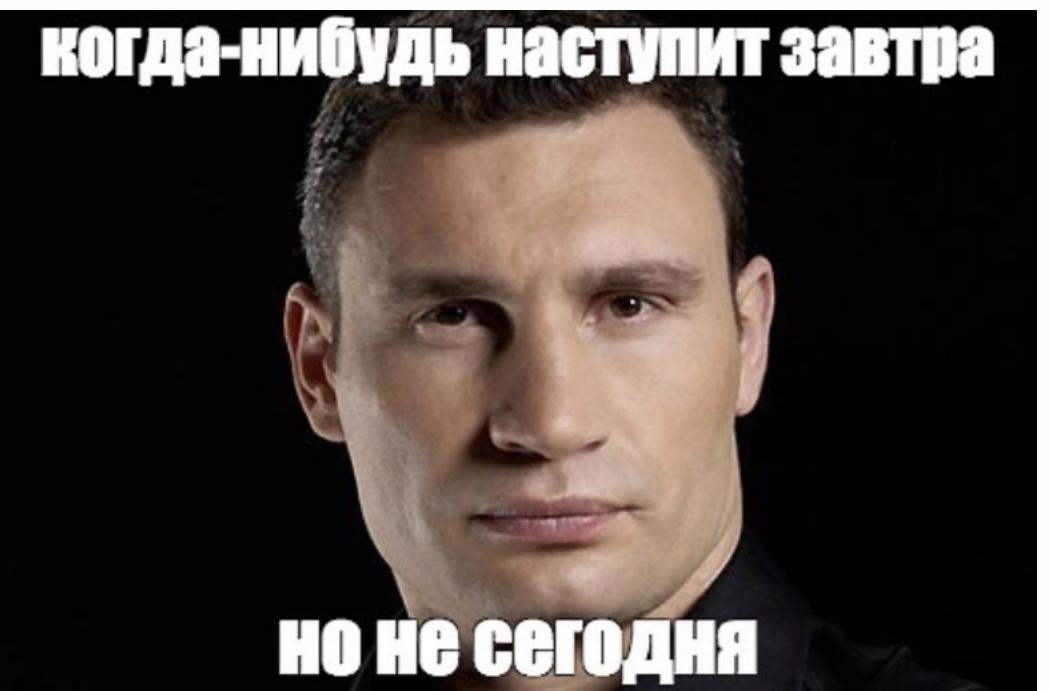


Дата и время

Что можно извлечь:

1. Абсолютное время
2. Периодичность(час, день, месяц...)
3. Временной интервал до особого события

когда-нибудь наступит завтра



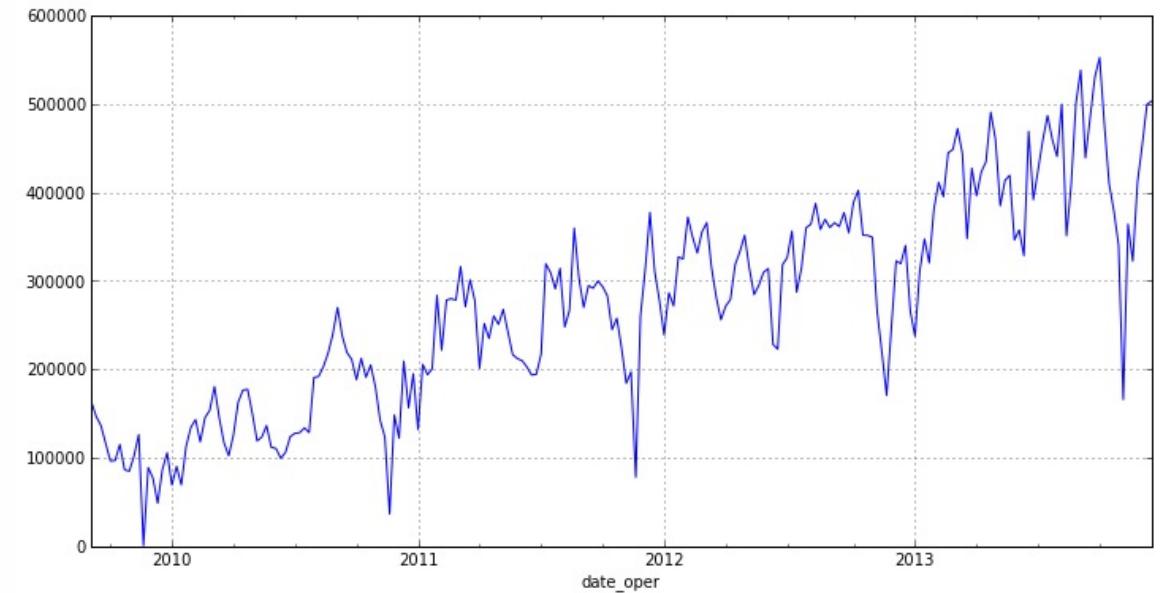
A black and white portrait of a man with short, dark hair, looking directly at the camera with a neutral expression. He is wearing a dark, collared shirt. The background is solid black.

но не сегодня

Временные ряды

Что можно извлечь:

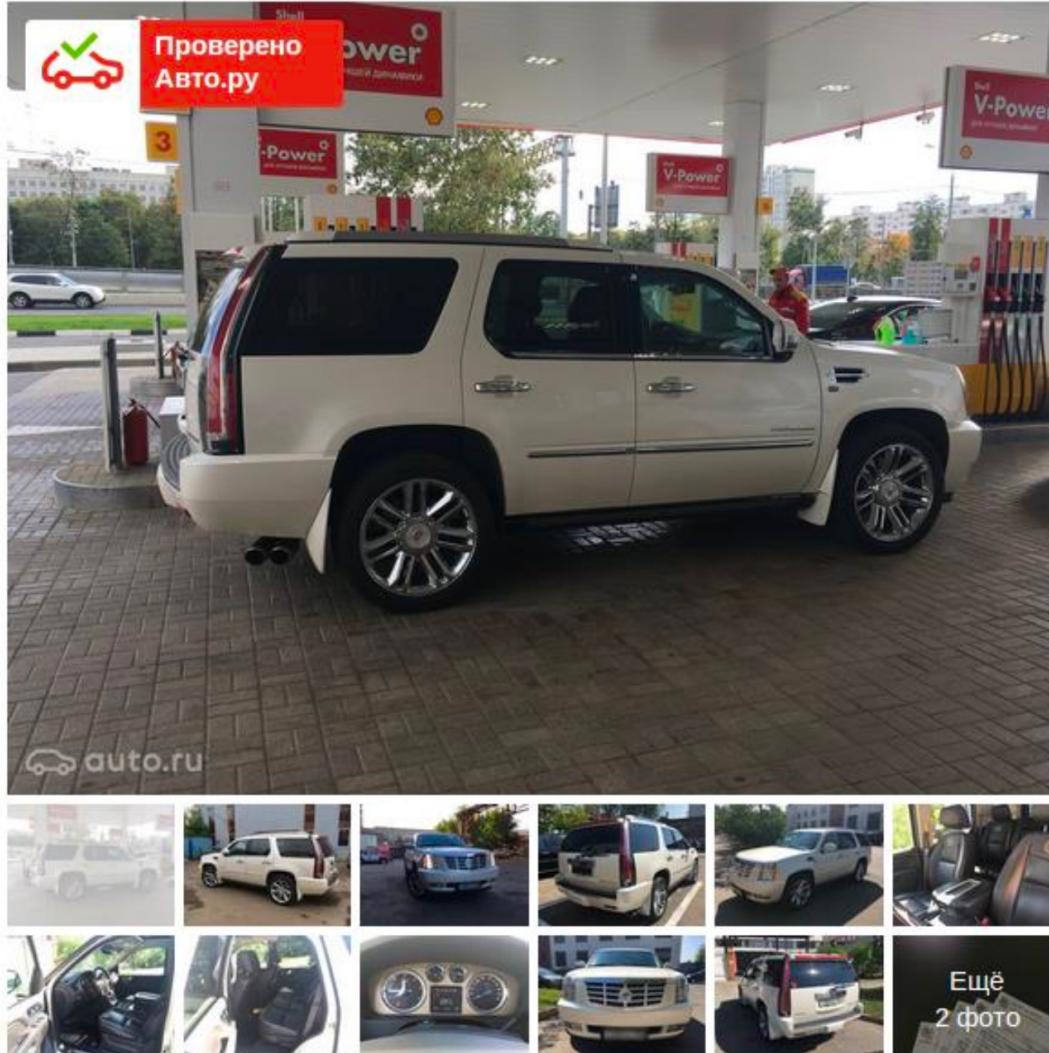
1. Среднее значение за период
2. Стандартное отклонение за период
3. Тренд за период
4. Количество пиков за период



Извлечение признаков

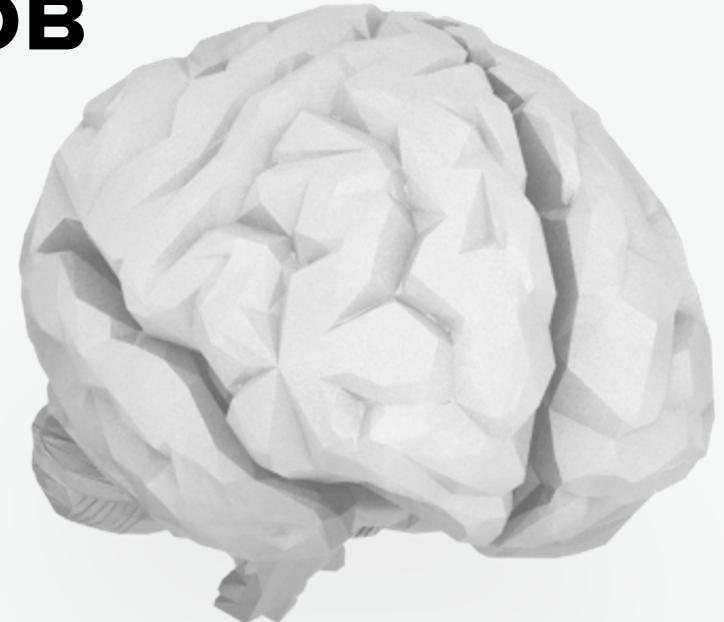
Год выпуска	2011
Пробег	98 000 км
Кузов	Внедорожник 5 дв.
Цвет	Белый
Двигатель	6.2 л / 409 л.с. / бензин
Коробка	Автоматическая
Привод	Полный
Руль	Левый
Состояние	Не требует ремонта
Владельцы	3 владельца
ПТС	Оригинал
Владение	9 месяцев
Таможня	Растаможен
VIN	XWFS47EF*C0****62
Автокод	Без ограничений

[Характеристики модели в каталоге](#)



Преобразование признаков

*Адаптируем признаковое представление
под конкретный тип модели.*



Зачем преобразовывать признаки?

1. Чтобы конкретный алгоритм машинного обучения их правильно интерпретировал
2. Чтобы конкретный алгоритм машинного обучения эффективно находил взаимосвязи
3. Чтобы внести априорные знания о наборе данных или свойствах признаков



Нормализация

Нормализация — это преобразование данных к неким безразмерным единицам. Иногда — в рамках заданного диапазона, например, [0..1] или [-1..1]. Иногда — с какими-то заданным свойством, как, например, стандартным отклонением равным 1.

Ключевая цель нормализации — приведение различных данных в самых разных единицах измерения и диапазонах значений к единому виду, который позволит сравнивать их между собой или использовать для расчёта схожести объектов. На практике это необходимо, например, для кластеризации и в некоторых алгоритмах машинного обучения.

Стандартизация

Для каждого признака в наборе вычитаем среднее и делим на стандартное отклонение.

Применяется к **количественным, порядковым и бинарным** признакам .

Актуально для:

Линейные модели

Метод ближайших соседей

Масштабирование

Значения каждого признака в наборе приводят к диапазону [0,1].

Применяется к **количественным, порядковым и бинарным** признакам .

Актуально для:

Линейные модели

Метод ближайших соседей

Монотонные преобразования

Применение монотонного преобразования к признаку
(например: логарифмирование, возведение в степень)

Применяется к **количественным и порядковым**
признакам .

Актуально для:
Линейные модели
Метод ближайших соседей

Бинаризация

Область значений **количественного или порядкового** признака делим на N участков и представляем в виде N бинарных признаков.

Применяется к **количественным и порядковым** признакам .

Актуально для:
Линейные модели

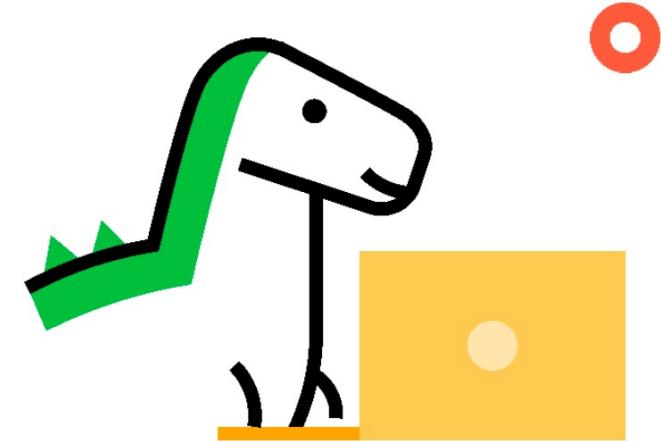
Преобразование признаков

Категориальные признаки

Label Encoding

Пример: имеется текстовое описание признаков

Не подходит для линейных моделей



✗

The diagram illustrates the process of transforming categorical features into numerical values. On the left, a table shows four rows of data with a 'Feature' column. An arrow points from this table to another table on the right, which contains the same data but with numerical values assigned to each category. The categories 'School', 'Basic', and 'University' are mapped to 1, 0, and 2 respectively, while 'School' appears again in the fourth row.

	Feature
1	School
2	Basic
3	University
4	School

→

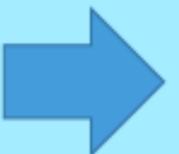
	Feature
1	1
2	0
3	2
4	1

Преобразование признаков

Категориальные признаки

One-Hot Encoding

Пример: имеется текстовое описание признаков



	Feature		F=School	F=Basic	F=University
1	School		1	0	0
2	Basic		0	1	0
3	University		0	0	1
4	School		1	0	0

Преобразование признаков

Категориальные признаки

Hashing trick

The diagram illustrates the 'Hashing trick' for transforming categorical features into numerical vectors. On the left, a table shows four rows of data with a 'Feature' column. A large blue arrow points from this table to a second table on the right, which represents the transformed vectors.

	Feature
1	School
2	Basic
3	University
4	School

A large blue arrow points from the left table to the right table.

	F=S	F=B,U
1	1	0
2	0	1
3	0	1
4	1	0

Задача

Алгоритм: k ближайших соседей с евклидовым расстоянием

Признаки:

1. категория кинотеатра [1..43]
2. день недели [0..6]
3. час суток [0..23]
4. цена билета [100..1000]

Целевая переменная: заполненность зала в % Что делать?

B jupyter notebook



Очистка данных



1) Удаление или преобразование пропущенных (неопределенных) данных - многие модели не допускают во входных данных пропуски



2) Удаление «нуль-вариантных» переменных (числовых и номинальных) - для многих моделей это может привести к краху или к нестабильной работе.



3) Выявление и удаление коррелированных предикторов (числовых) - некоторые модели отлично справляются с коррелированными предикторами (например PLS, LARS и подобные, использующие L1 регуляризацию), другие модели могут получить преимущества от снижения уровня корреляции между предикторами.

Пропуски



Большинство реальных данных имеют пропущенные значения:

- Ошибки при записи
- Ошибки при измерении
- Невозможность сбора

Далеко не все алгоритмы умеют работать с неполными данными

Заполнение пропусков

- Заменять наиболее вероятным – в случае непрерывных данных замена на среднее значение из наиболее вероятного интервала; в дискретном случае – выбирается значение с наибольшей вероятностью.
- Заменять случайными значениями – замена пропусков на случайное значение из распределения.
- Заменять средним/медианой
- Заменять значением Не задано – доступно только для дискретного поля, выполняется замена пропусков на значение «Не задано».
- Удалять записи – строки с выявленными пропусками исключаются из набора данных. Метод недоступен для упорядоченных рядов. **Ничего не испортим, но что если данных и так мало?**
- Заполняем прогнозным значением

Отбор признаков (feature selection)

Отбор признаков – это выбор признаков, имеющих наиболее тесные взаимосвязи с целевой переменной.

Обеспечивает три основных преимущества:

1. **Уменьшение переобучения.** Чем меньше избыточных данных, тем меньше возможностей для модели принимать решения на основе «шума».
2. **Повышение точности.** Чем меньше противоречивых данных, тем выше точность.
3. **Сокращение времени обучения.** Чем меньше данных, тем быстрее обучаются модель.

Методы отбора признаков

1. Одномерный отбор признаков

Отбор признаков по взаимосвязи с целевой переменной, могут быть отобраны с помощью статистических критериев (например, хи-квадрат).

2. Рекурсивное исключение признаков

Метод рекурсивного исключения признаков (recursive feature elimination, RFE)

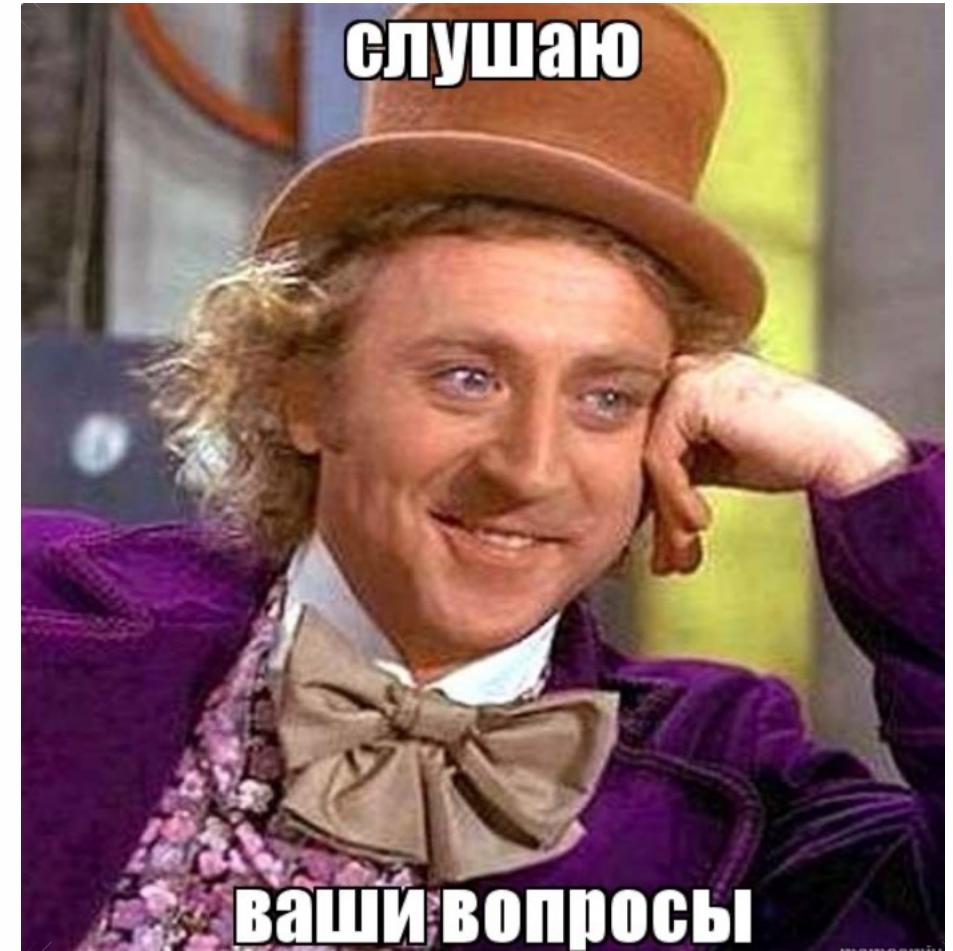
3. Метод главных компонент

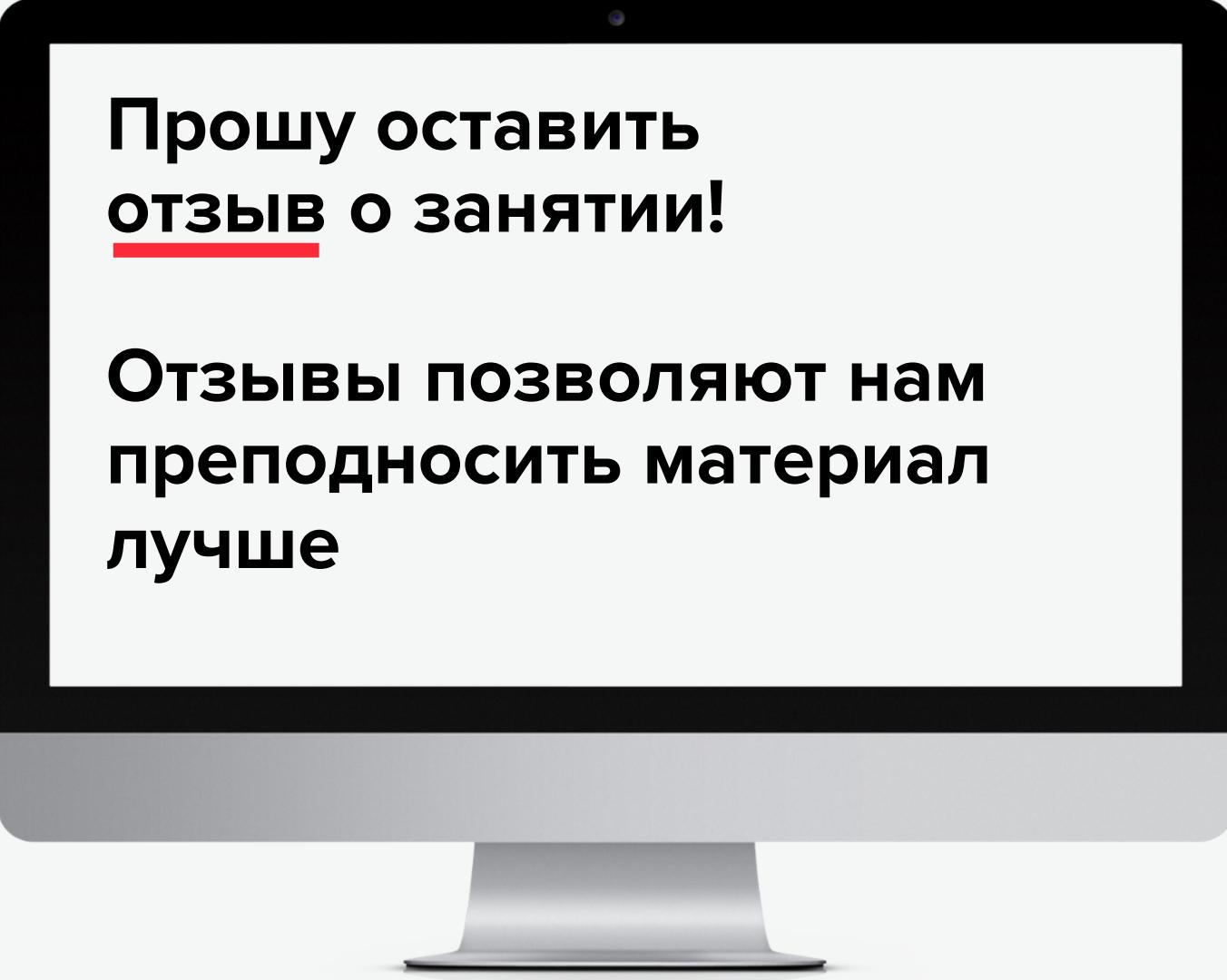
Метод главных компонент (principal component analysis, PCA) позволяет уменьшить размерность данных с помощью преобразования на основе линейной алгебры

4. Отбор на основе важности признаков

Ансамблевые алгоритмы на основе деревьев решений, такие как случайный лес (random forest), позволяют оценить важность признаков

Вопросы?





**Прошу оставить
отзыв о занятии!**

**Отзывы позволяют нам
преподносить материал
лучше**

**СПАСИБО
ЗА ВНИМАНИЕ**

