Интерпретируемое машинное обучение

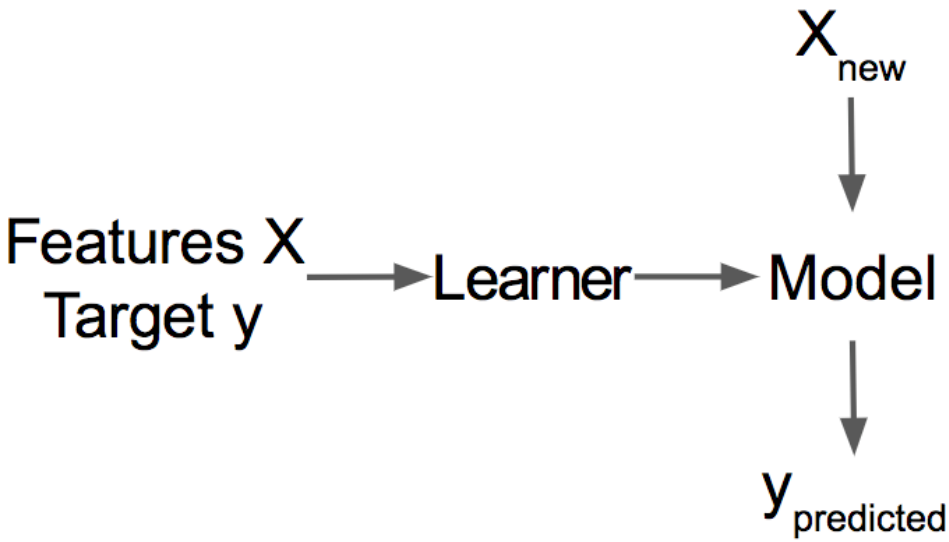
Студент: Тыцкий В.И.

Научный руководитель: Майсурадзе А.И.

МГУ имени М. В. Ломоносова, факультет ВМК, кафедра ММП

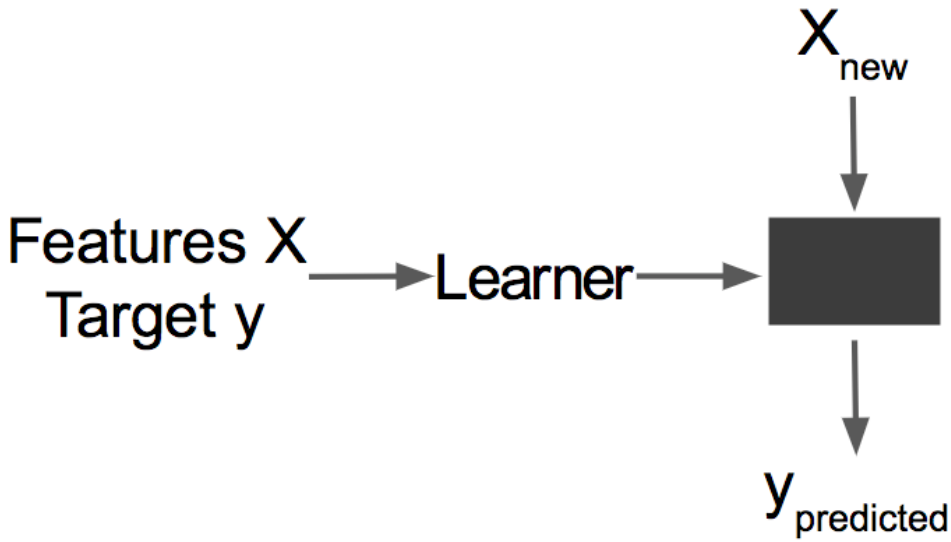
Оглавление

Данные ⇒ Модель ⇒ Done?



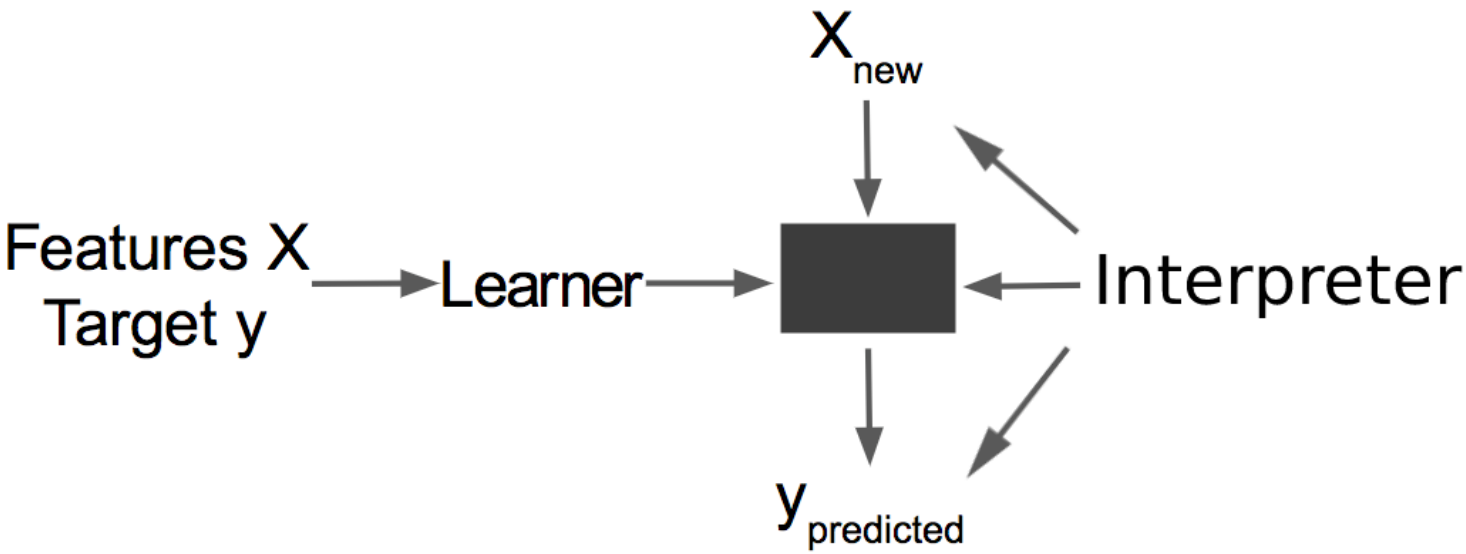
- Постановка задачи
- Определились, что будем решать с помощью ML
- Сбор данных
- Обучили модель, она работает "хорошо"
- На этом можно остановиться?

Проблема черного ящика



- Сбор данных и постановка задачи
- Определились, что будем решать с помощью ML
- Обучили модель, она работает "хорошо"
- **Вместо модели получили "умный" черный ящик**

Почему мы можем тебе верить?



- Почему модель приняла именно такое решение?
- На что модель обращает внимание?
- Какие свойства объекта наиболее важны в общем и в совокупности

Хорошего качества недостаточно

- Использование ML несет за собой большие риски:
  - Медицина
  - Государственные структуры
  - Банки
- Глубже понимаем наблюдаемое явление
- Новые закономерности в данных
- Уверенность в адекватности модели

Классификация методов интерпретации

- Локальные – объясняют модель на конкретном объекте
- Глобальные – объясняют как модель работает в целом

- Специфичные для модели (Model Specific)
- Индифферентные к модели (Model Agnostic)

Model = Interpreter

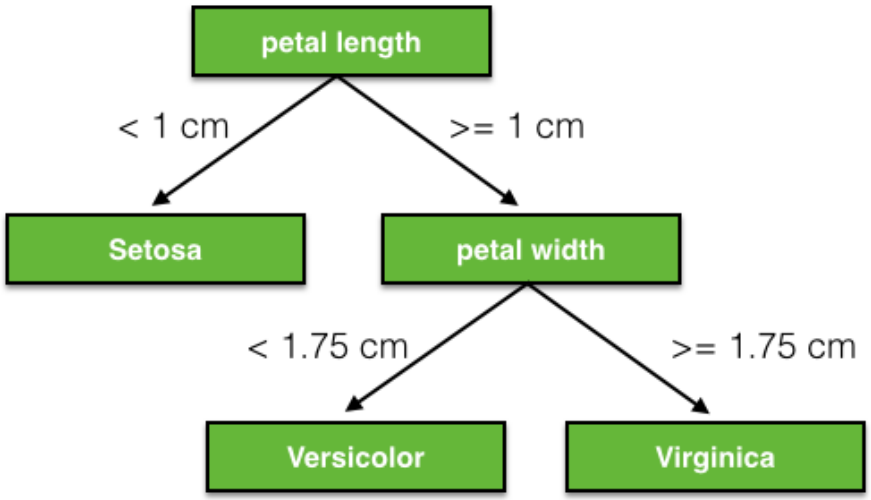
- Используем простые модели: линейные, деревья
- Структура построения алгоритма сообщает принцип принятия решений
- Недостаток: не всегда справляются с задачей

Model = Interpreter

В случае линейных моделей можем посмотреть на вес для каждого признака

$$y_i = f(< \vec{w}, \vec{x_i} >)$$

В случае решающего дерева



LIME (Local interpretable model-agnostic explanations)

- x - конкретный объект
- f - черный ящик (мы умеем $x \to f(x) \to y$)
- g - простая модель (например линейная)
- π_x - насколько сильно учитываем контекст вокруг
- $\Omega(g)$ - мера сложности модели g

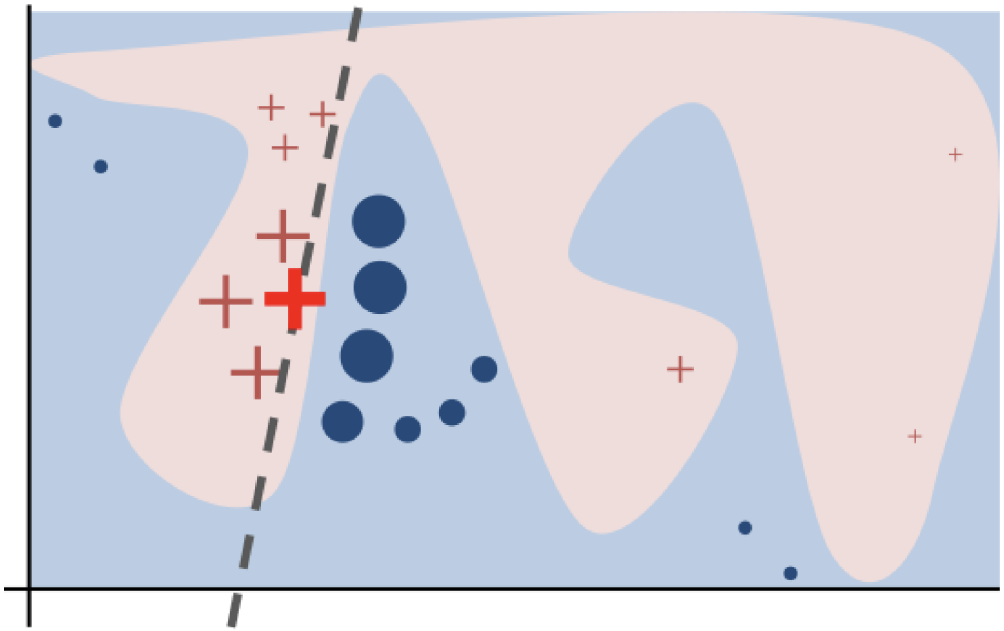
$$interpreter(x) = \underset{g}{\operatorname{argmin}}\, L(f, g, \pi_x) + \Omega(g)$$

LIME

$$interpreter(x) = \underset{g}{\operatorname{argmin}}\, L(f, g, \pi_x) + \Omega(g)$$

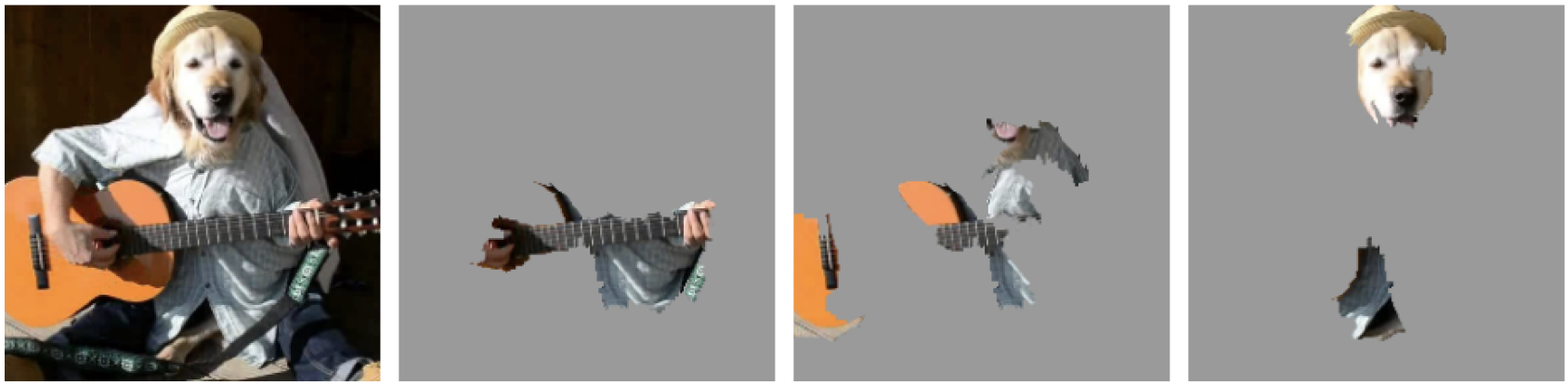
$$L(f, g, \pi_x) = \sum_{x'} (f(x') - g(x'))^2 \pi_x(x, x')$$

Объясняем объясняющего



1. В качестве простой модели возьмем **взвешенную** линейную
2. Выберем объект, который хотим проанализировать
3. Обучаем линейную регрессию в окрестности объекта
4. Интерпретируем линейную регрессию!

Изображения



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

- Для изображений сложно сказать, что такое окрестность
- Будем находить суперпиксели и менять их

Shapley values

- Концепция из кооперативной теории игр
- Считаем, что награда пропорциональна вкладу игрока

P – все игроки, R – всевозможные перестановки игроков
$u(K)$ – награда множества игроков K
P_i^R – игроки, встретившиеся в перестановке R до i-ого игрока

$$\phi_i(u) = \frac{1}{N!} \sum_R \left[u(P_i^R \cup \{i\}) - u(P_i^R) \right]$$

Shapley values

- Линейность – $\phi_i(u+v) = \phi_i(u) + \phi_i(v)$
- Симметричность – награда игрока не зависит от его номера
- Аксиома Болвана – бесполезный игрок не вносит вклад в коалицию
- Эффективность – $\sum_i \phi_i(u) = u(P)$

$$\phi_i(u) = \frac{1}{N!} \sum_{S \subseteq P \setminus \{i\}} \frac{1}{C_N^{|S|}} \left[u(S \cup \{i\}) - u(S) \right]$$

SHAP[1]

Игрок $\rightarrow$ признак
Награда $\rightarrow$ значение $f(x)$

- Выбираем $x \in R^D$
- Семплируем "подмножества признаков" $z'_k \in \{0,1\}^D, k \in \{1,\dots,K\}$
- Для каждого z'_k **восстанавливаем** $z_k = h_x(z'_k)$ и находим $f(h_x(z'_k))$
- Считаем веса для каждого z'_k и обучаем линейную модель
- Веса линейной модели будут обладать свойствами shapley values

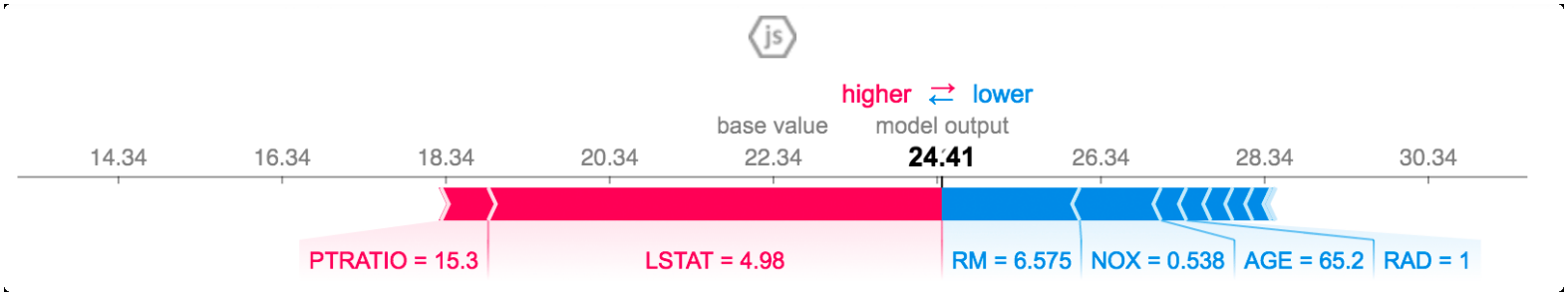
[1] Некоторые теоретические моменты опущены

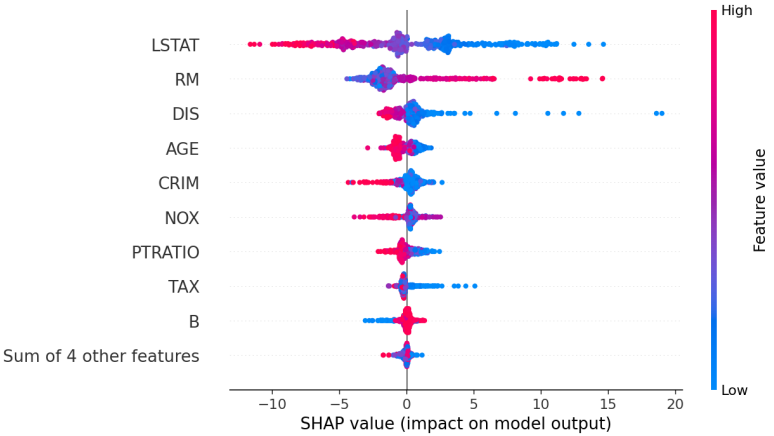
SHAP

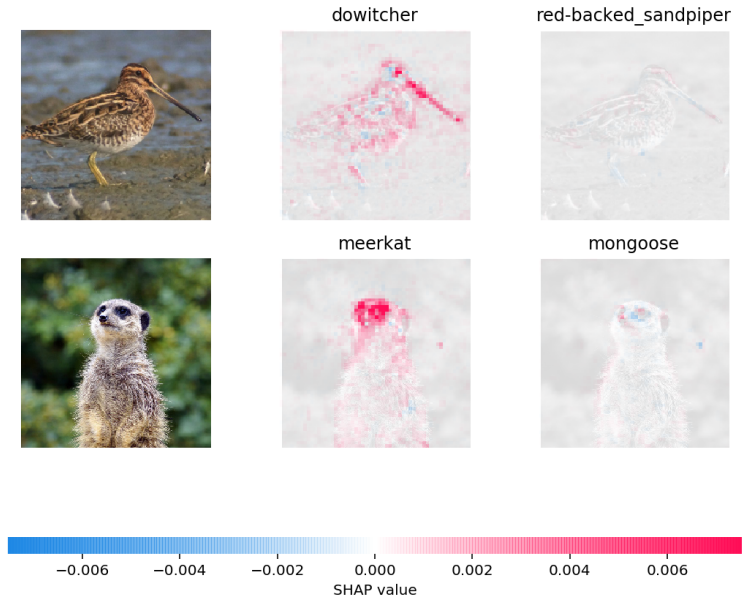
- Операция восстановления $h_x : \{0,1\}^D \to R^D$ "заменяет" 0 на случайное значение признака из набора данных, а 1 на значение признака у рассматриваемого x
- Вес для z'_i считаем по следующей формуле:

$$\pi_x(z') = \frac{(M-1)}{C_D^{|z'|}(M-|z'|)|z'|}$$

SHAP


higher ⇄ lower
base value
model output
14.34
16.34
18.34
20.34
22.34
24.41
26.34
28.34
30.34
PTRATIO = 15.3
LSTAT = 4.98
RM = 6.575
NOX = 0.538
AGE = 65.2
RAD = 1


LSTAT
RM
DIS
AGE
CRIM
NOX
PTRATIO
TAX
B
Sum of 4 other features
−10
−5
0
5
10
15
20
SHAP value (impact on model output)
High
Feature value
Low


dowitcher
red-backed_sandpiper
meerkat
mongoose
−0.006
−0.004
−0.002
0.000
0.002
0.004
0.006
SHAP value

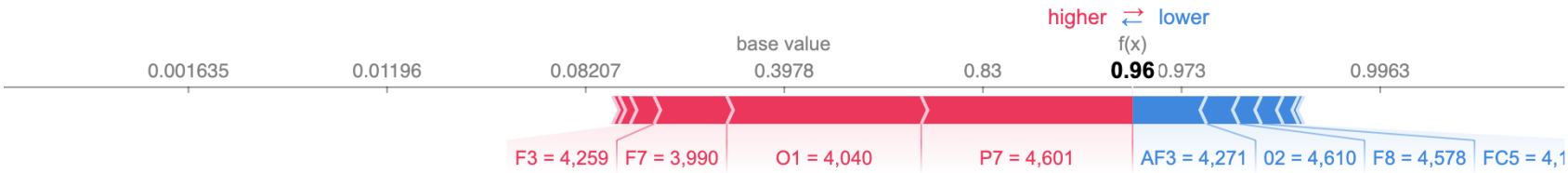
Проблемы временных рядов для интерпретации

- Длина последовательности бывает разной
- Огромная размерность пространства
- Один timestamp редко несет в себе много информации
- Чем больше признаковое пространство, тем сложнее разобраться

Задачи seq2seq

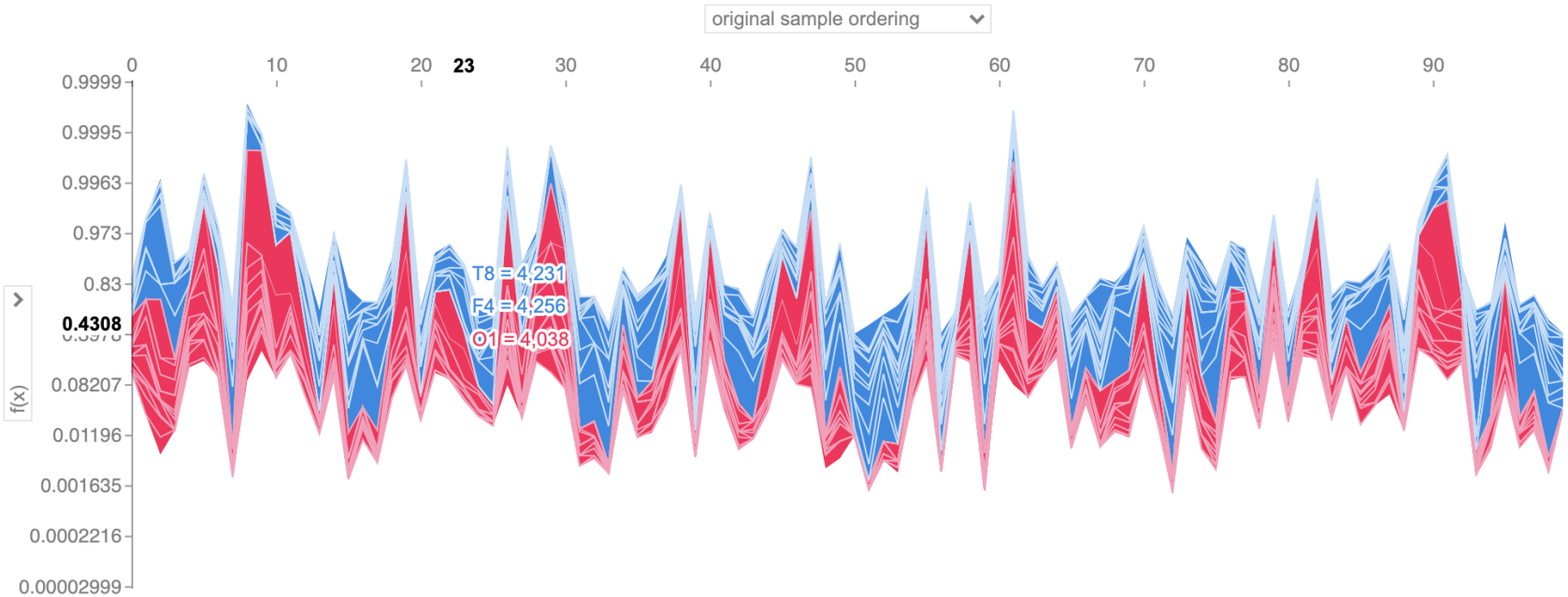
- Пусть дана задача seq2seq (открыты ли глаза у человека в конкретный момент)
- Для простоты будем считать, что каждый промежутки времени независимы между собой
- Обучим модель
- Попробуем применить один из методов объяснения модели

Задачи seq2seq



- Accuracy $= 0.88$, AUC-ROC $= 0.94$

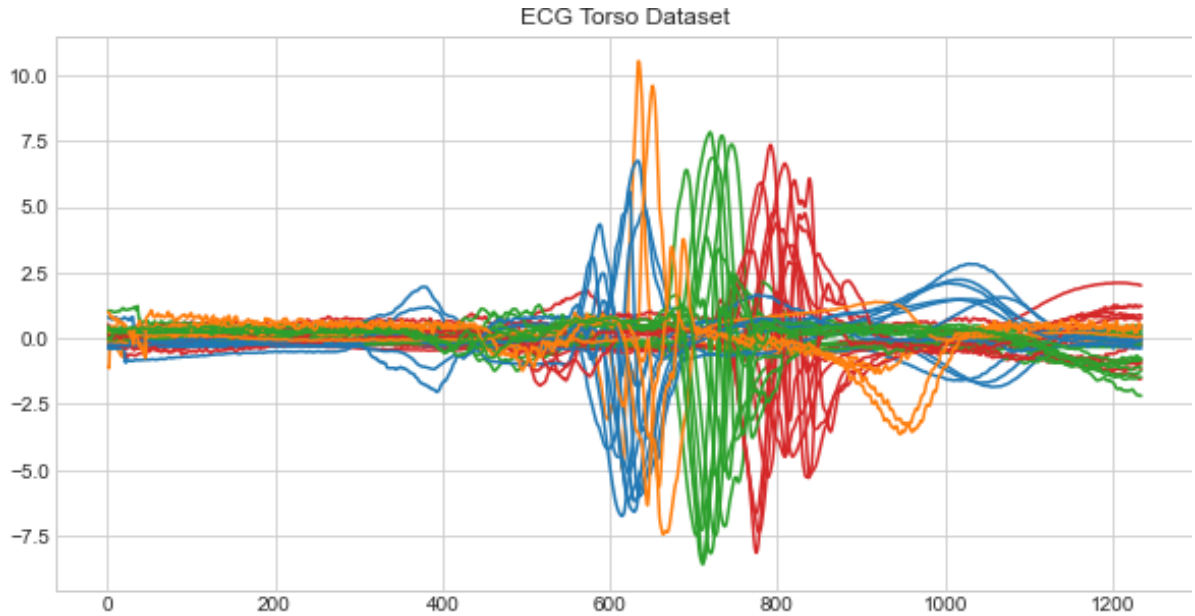
Задачи seq2seq



Как сделать лучше?

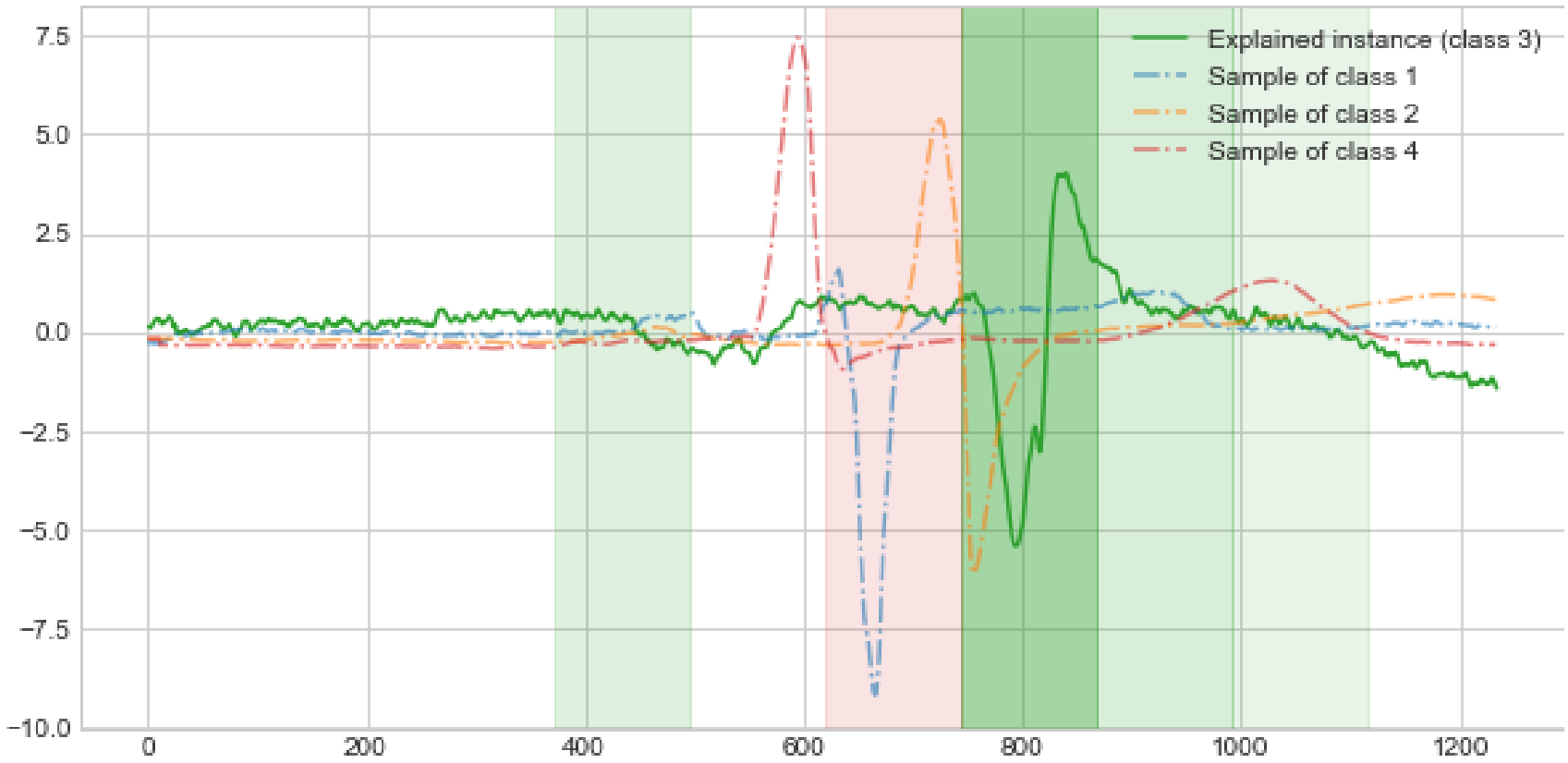
- В качестве признаков на текущем моменте брать несколько состояний предыдущих моментов
- Использовать RNN и считать hidden state некоторым "суперпризнаком"
- Для нахождения нетривиальных зависимостей можно использовать графовые нейросети

Задачи seq2label

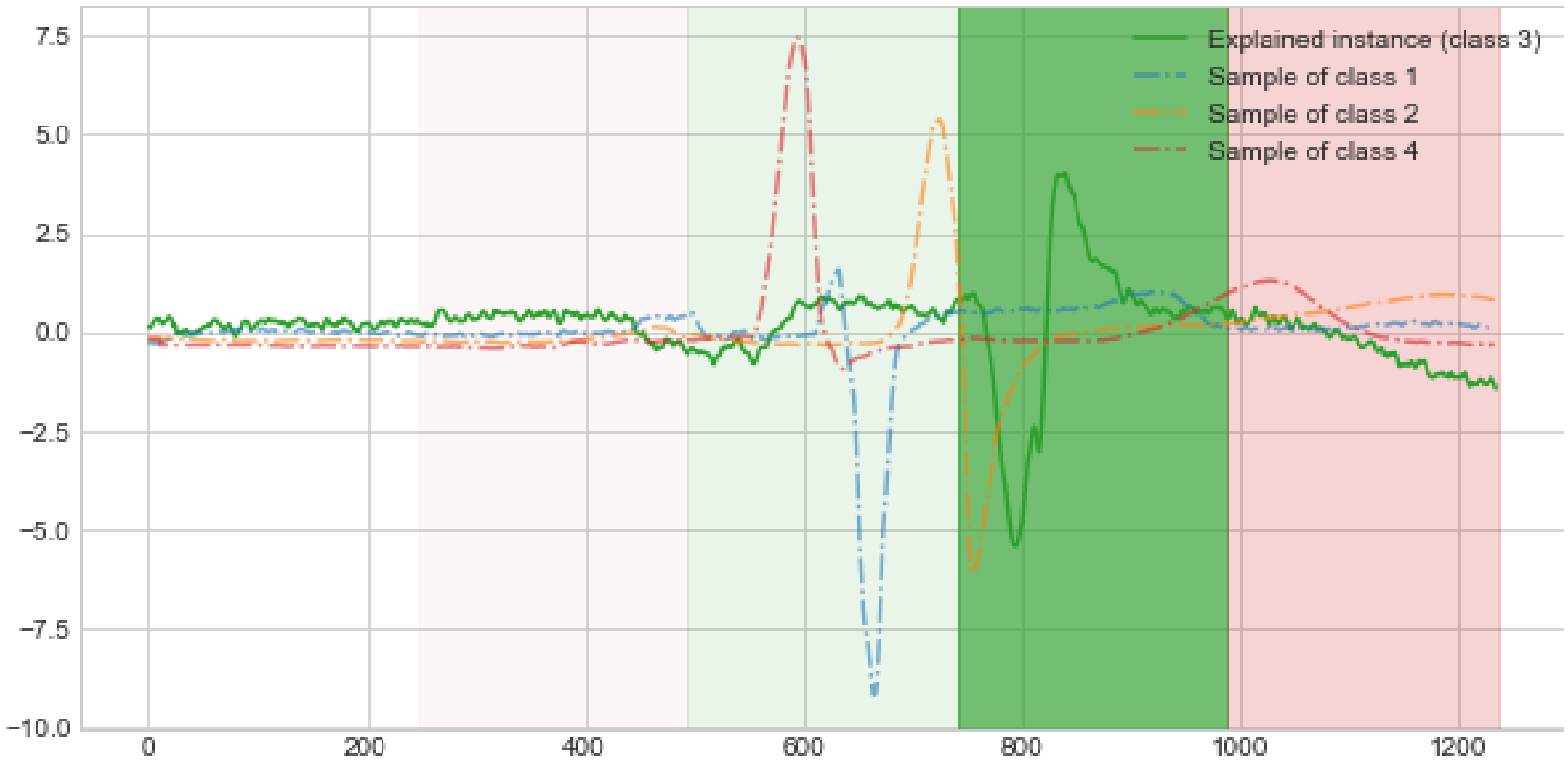


- Рассмотрим задачу seq2label (распознавание человека по его ЭКГ)
- Разделим временной ряд на равные кусочки и будем считать их "суперпризнаками"
- Выключение "суперпризнака" = замена на среднее/шум

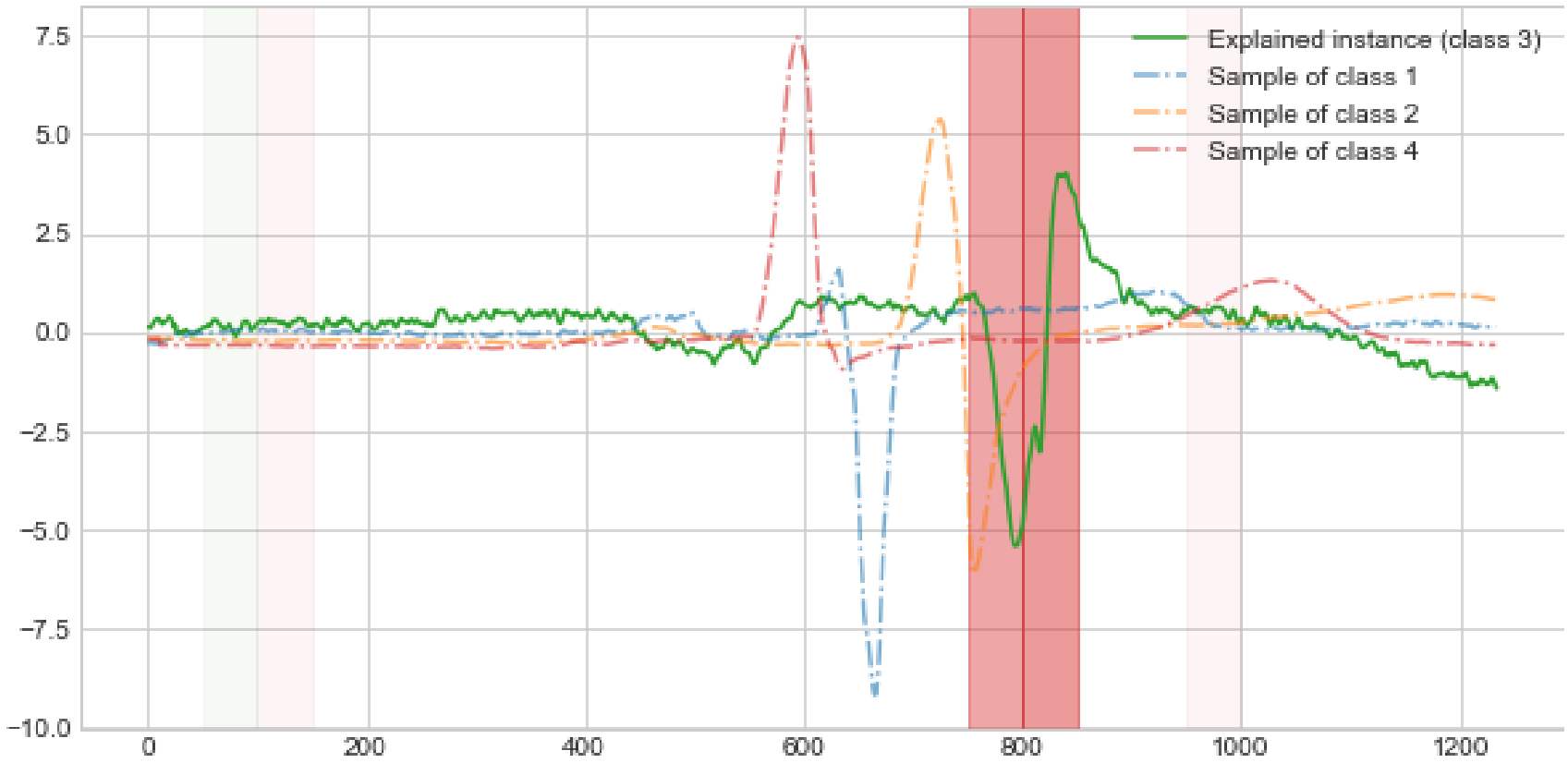
Задачи seq2label: 15 slices



Задачи seq2label: 5 slices



Задачи seq2label: 25 slices



Дальнейшие исследования

- Сверточные нейронные сети
- Графовые нейронные сети
- Реккурентные нейронные сети
- Использование SHAP вместо LIME для задачи seq2label

Спасибо за внимание!

Библиография

1. Christoph Molnar. Interpretable Machine Learning.
2. Lundberg S. M., Lee S. I. A unified approach to interpreting model predictions – 2017. – С. 4768-4777.
3. Ribeiro M. T., Singh S., Guestrin C. "Why should i trust you?" Explaining the predictions of any classifier – 2016. – С. 1135-1144.
4. André Ferreira. Interpreting recurrent neural networks on multivariate time series
5. CinCECGtorso
6. EEG Eye Blinking Prediction