

# Методы интерпретации моделей для задач NLP

Колесников Павел  
ВШЭ ФКН, группа БПМИ-193

22 февраля 2022 г.

# План доклада

---

1. Зачем нам интерпретировать?

2. Что влияет на выбор метода интерпретации?

3. Black-box методы

3.1 Partial Dependence Plot (PDP)

3.2 Individual Conditional Expectation (ICE)

3.3 Permuted Feature Importance (PFI)

3.4 Local Interpretable Model-agnostic Explanations (LIME)

3.5 SHapley Additive exPlanation values (SHAP values)

4. White-box методы

4.1 Gradient-based подход к интерпретации

Token attribution

Integrated Grads

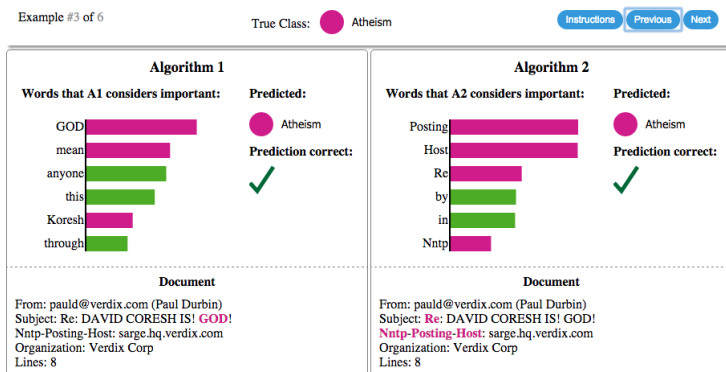
4.2 Визуализация attention слоев

5. Как оценить качество интерпретации?

6. Готовые решения для интерпретации

# Зачем нам интерпретировать?

- Для решения практических задач (например, в банковской среде)
- Чтобы проверить, что модель разумная



# Что влияет на выбор метода интерпретации?

---

- Тип модели
  - Простые модели: линейные, решающие деревья
  - Сложные модели: нейронные сети, ансамбли

# Что влияет на выбор метода интерпретации?

---

- **Тип модели**
  - Простые модели: линейные, решающие деревья
  - Сложные модели: нейронные сети, ансамбли
- **Что хотим понять?**
  - Всю логику модели
  - Решение модели на конкретном примере

# Что влияет на выбор метода интерпретации?

---

- **Тип модели**
  - Простые модели: линейные, решающие деревья
  - Сложные модели: нейронные сети, ансамбли
- **Что хотим понять?**
  - Всю логику модели
  - Решение модели на конкретном примере
- **К чему имеем доступ?**
  - Только к данным и модели-черному ящику
  - К данным и внутреннему устройству модели

# Black-box методы



# Partial Dependence Plot (PDP)

---

## Идея:

Давайте зафиксируем какой-то параметр и усредним предсказание модели по всем объектам, затем построим график зависимости предсказания от зафиксированного параметра.



# Partial Dependence Plot (PDP)

---

**Формально:**

$\hat{F}$  - модель,  $x = \{x_1, x_2, \dots, x_n\}$  - вход модели

# Partial Dependence Plot (PDP)

---

**Формально:**

$\hat{F}$  - модель,  $x = \{x_1, x_2, \dots, x_n\}$  - вход модели

$z_I = \{z_1, z_2, \dots, z_I\} \subset \{x_1, x_2, \dots, x_n\}$ ,  $z_{\setminus I} = x \setminus z_I$

# Partial Dependence Plot (PDP)

---

**Формально:**

$\hat{F}$  - модель,  $x = \{x_1, x_2, \dots, x_n\}$  - вход модели

$z_I = \{z_1, z_2, \dots, z_I\} \subset \{x_1, x_2, \dots, x_n\}$ ,  $z_{\setminus I} = x \setminus z_I$

$\hat{F}(x) = \hat{F}(z_I, z_{\setminus I})$

# Partial Dependence Plot (PDP)

---

**Формально:**

$\hat{F}$  - модель,  $x = \{x_1, x_2, \dots, x_n\}$  - вход модели

$z_I = \{z_1, z_2, \dots, z_I\} \subset \{x_1, x_2, \dots, x_n\}$ ,  $z_{\setminus I} = x \setminus z_I$

$\hat{F}(x) = \hat{F}(z_I, z_{\setminus I})$

$\bar{Z}_I(z_I) = \int \hat{Z}(z_I, z_{\setminus I}) p(z_{\setminus I}) dz_{\setminus I}$ , где  $p(z_{\setminus I}) = \int p(x) dz_I$

# Partial Dependence Plot (PDP)

---

**Формально:**

$\hat{F}$  - модель,  $x = \{x_1, x_2, \dots, x_n\}$  - вход модели

$z_I = \{z_1, z_2, \dots, z_I\} \subset \{x_1, x_2, \dots, x_n\}$ ,  $z_{\setminus I} = x \setminus z_I$

$\hat{F}(x) = \hat{F}(z_I, z_{\setminus I})$

$\bar{Z}_I(z_I) = \int \hat{Z}(z_I, z_{\setminus I}) p(z_{\setminus I}) dz_{\setminus I}$ , где  $p(z_{\setminus I}) = \int p(x) dz_I$

Для какой-то конкретной выборки эта функция может быть оценена:

$\bar{Z}_I(z_I) \approx \frac{1}{N} \sum_{i=1}^N \hat{Z}(z_I, z_{\setminus I})$

# Partial Dependence Plot (PDP)

---

**Формально:**

$\hat{F}$  - модель,  $x = \{x_1, x_2, \dots, x_n\}$  - вход модели

$z_I = \{z_1, z_2, \dots, z_I\} \subset \{x_1, x_2, \dots, x_n\}$ ,  $z_{\setminus I} = x \setminus z_I$

$\hat{F}(x) = \hat{F}(z_I, z_{\setminus I})$

$\bar{Z}_I(z_I) = \int \hat{Z}(z_I, z_{\setminus I}) p(z_{\setminus I}) dz_{\setminus I}$ , где  $p(z_{\setminus I}) = \int p(x) dz_I$

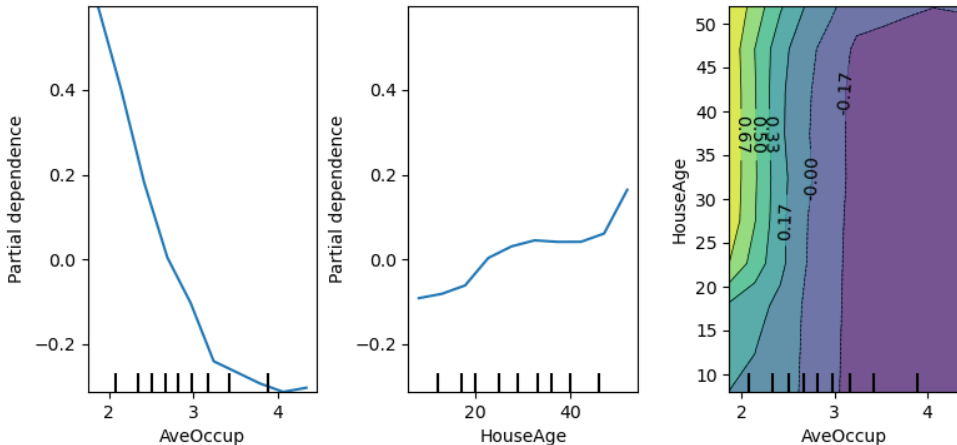
Для какой-то конкретной выборки эта функция может быть оценена:

$\bar{Z}_I(z_I) \approx \frac{1}{N} \sum_{i=1}^N \hat{Z}(z_I, z_{\setminus I})$

PDP - график  $\bar{Z}_I(z_I)$

# Примеры PDP

Partial dependence of house value on non-location features  
for the California housing dataset, with Gradient Boosting



# Достоинства и недостатки PDP

---

## Плюсы:

- + Не зависит от внутреннего устройства модели
- + Интуитивен
- + Легок в реализации
- + Показывает глобальные зависимости

## Минусы:

- Предполагаем независимость признаков
- Гетерогенные эффекты могут быть скрыты
- Не очень понятно, как использовать в задачах NLP



# Individual Conditional Expectation (ICE)

---

**Идея:**

Берем PDP, но рисуем график не для усредненной функции, а для всех объектов выборки.

# Individual Conditional Expectation (ICE)

---

**Идея:**

Берем PDP, но рисуем график не для усредненной функции, а для всех объектов выборки.

**Формально:**

# Individual Conditional Expectation (ICE)

---

**Идея:**

Берем PDP, но рисуем график не для усредненной функции, а для всех объектов выборки.

**Формально:**

$\hat{F}$  - модель,  $x_i = \{x_{1,i}, x_{2,i}, \dots, x_{n,i}\}$  - вход модели

# Individual Conditional Expectation (ICE)

---

## Идея:

Берем PDP, но рисуем график не для усредненной функции, а для всех объектов выборки.

## Формально:

$\hat{F}$  - модель,  $x_i = \{x_{1,i}, x_{2,i}, \dots, x_{n,i}\}$  - вход модели

$z_I = \{z_1, z_2, \dots, z_I\} \subset \{x_1, x_2, \dots, x_n\}$

# Individual Conditional Expectation (ICE)

---

## Идея:

Берем PDP, но рисуем график не для усредненной функции, а для всех объектов выборки.

## Формально:

$\hat{F}$  - модель,  $x_i = \{x_{1,i}, x_{2,i}, \dots, x_{n,i}\}$  - вход модели

$z_l = \{z_1, z_2, \dots, z_l\} \subset \{x_1, x_2, \dots, x_n\}$

$\bar{Z}_{l,i}(z_l) = \hat{Z}(z_l, x_i \setminus z_l)$

# Individual Conditional Expectation (ICE)

---

## Идея:

Берем PDP, но рисуем график не для усредненной функции, а для всех объектов выборки.

## Формально:

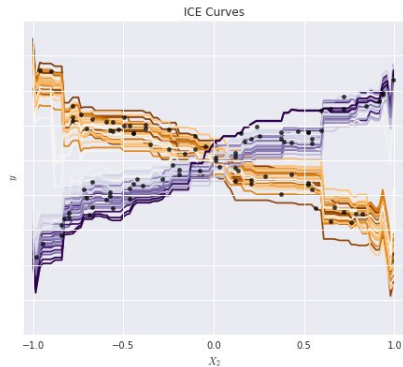
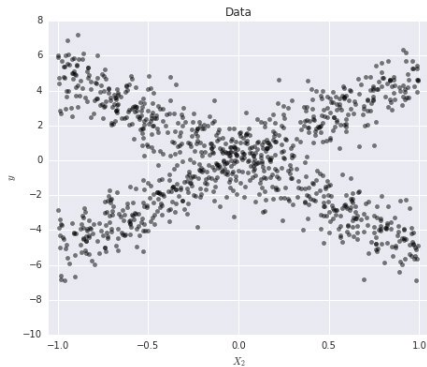
$\hat{F}$  - модель,  $x_i = \{x_{1,i}, x_{2,i}, \dots, x_{n,i}\}$  - вход модели

$z_l = \{z_1, z_2, \dots, z_l\} \subset \{x_1, x_2, \dots, x_n\}$

$\bar{Z}_{l,i}(z_l) = \hat{Z}(z_l, x_i \setminus z_l)$

ICE - график  $\bar{Z}_{l,i}(z_l)$  для всех  $i$

# Примеры ICE



# Достоинства и недостатки ICE

---

## Плюсы:

- + Не зависит от внутреннего устройства модели
- + Интуитивен
- + Легок в реализации
- + Показывает глобальные зависимости
- + Позволяет избежать влияния гетерогенности выборки

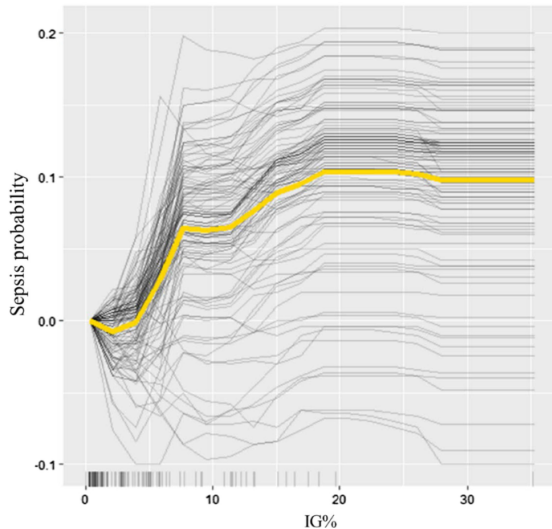
## Минусы:

- Не позволяет увидеть влияние более чем одного признака
- Предполагаем независимость признаков
- Не очень просто увидеть усредненный тренд
- Не очень понятно, как использовать в задачах NLP



# Комбинация PDP и ICE

---



# Permuted Feature Importance (PFI)

---

Идея:

Давайте вместо какого-то признака будет использовать значение этого признака у случайного объекта нашей выборки и смотреть как поменялось качество модели.

$s_{k,i}$  - качество модели на выборке  $X_{k,i}$

$$PFI_j = s - \frac{1}{K} \sum_{i=1}^K s_{j,i}$$

# Permuted Feature Importance (PFI)

---

**Идея:**

Давайте вместо какого-то признака будет использовать значение этого признака у случайного объекта нашей выборки и смотреть как поменялось качество модели.

**Формально:**

$\hat{F}$  - модель,  $X$  - выборка,  $s$  - качество модели на выборке  $X$

$s_{k,i}$  - качество модели на выборке  $X_{k,i}$

$$PFI_j = s - \frac{1}{K} \sum_{i=1}^K s_{j,i}$$

# Permuted Feature Importance (PFI)

---

**Идея:**

Давайте вместо какого-то признака будет использовать значение этого признака у случайного объекта нашей выборки и посмотреть как поменялось качество модели.

**Формально:**

$\hat{F}$  - модель,  $X$  - выборка,  $s$  - качество модели на выборке  $X$

$X_{k,i}$  - выборка, в которой  $k$ -ый признак перемешан между всеми объектами выборки.

$s_{k,i}$  - качество модели на выборке  $X_{k,i}$

$$PFI_j = s - \frac{1}{K} \sum_{i=1}^K s_{j,i}$$

# Permuted Feature Importance (PFI)

---

**Идея:**

Давайте вместо какого-то признака будет использовать значение этого признака у случайного объекта нашей выборки и смотреть как поменялось качество модели.

**Формально:**

$\hat{F}$  - модель,  $X$  - выборка,  $s$  - качество модели на выборке  $X$

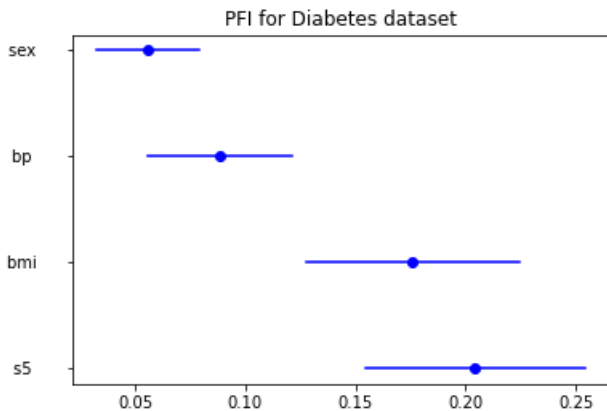
$X_{k,i}$  - выборка, в которой  $k$ -ый признак перемешан между всеми объектами выборки.

$s_{k,i}$  - качество модели на выборке  $X_{k,i}$

$$PFI_j = s - \frac{1}{K} \sum_{i=1}^K s_{j,i}$$

# Пример PFI

---



# Достоинства и недостатки PFI

---

## Плюсы:

- + В крайне сжатом формате показывает глобальные зависимости в модели
- + Можно адаптировать для задач NLP
- + Позволяет сравнивать результаты для разных функций потерь (если использовать относительный PFI)
- + Позволяет сравнивать друг с другом важности признаков

## Минусы:

- Предполагаем независимость признаков
- Вносим элемент случайности в метрику
- Не аддитивна (при увеличении PFI качество модели не обязательно падает)

# Local Interpretable Model-agnostic Explanations (LIME)

---

**Идея:** Давайте обучим модель, которую легко интерпретировать, предсказывать вывод сложной модели на примере и его небольших преобразованиях, а затем выделим те признаки предмета, которые сильнее влияют на предсказание простой модели.



# Local Interpretable Model-agnostic Explanations (LIME)

---

Формально:

# Local Interpretable Model-agnostic Explanations (LIME)

---

**Формально:**

$\hat{F}$  - модель,  $G$  - семейство интерпретируемых алгоритмов,  $g \in G$ ,  $x$  - объект из выборки,  $x'$  - интерпретируемая репрезентация  $x$ ,  $\Pi_x(z)$  - мера близости  $z$  и  $x$ .

# Local Interpretable Model-agnostic Explanations (LIME)

---

**Формально:**

$\hat{F}$  - модель,  $G$  - семейство интерпретируемых алгоритмов,  $g \in G$ ,  $x$  - объект из выборки,  $x'$  - интерпретируемая репрезентация  $x$ ,  $\Pi_x(z)$  - мера близости  $z$  и  $x$ .  
 $\mathcal{L}(\hat{F}, g, \Pi_x(z))$  - критерий качества аппроксимации  $\hat{F}$  функцией  $g$  в окрестности  $x$  определяемой  $\Pi_x(z)$

# Local Interpretable Model-agnostic Explanations (LIME)

---

**Формально:**

$\hat{F}$  - модель,  $G$  - семейство интерпретируемых алгоритмов,  $g \in G$ ,  $x$  - объект из выборки,  $x'$  - интерпретируемая репрезентация  $x$ ,  $\Pi_x(z)$  - мера близости  $z$  и  $x$ .  
 $\mathcal{L}(\hat{F}, g, \Pi_x(z))$  - критерий качества аппроксимации  $\hat{F}$  функцией  $g$  в окрестности  $x$  определяемой  $\Pi_x(z)$   
 $\Omega(g)$  - мера сложности интерпретации функции  $g$ .

# Local Interpretable Model-agnostic Explanations (LIME)

---

Формально:

$\hat{F}$  - модель,  $G$  - семейство интерпретируемых алгоритмов,  $g \in G$ ,  $x$  - объект из выборки,  $x'$  - интерпретируемая репрезентация  $x$ ,  $\Pi_x(z)$  - мера близости  $z$  и  $x$ .  
 $\mathcal{L}(\hat{F}, g, \Pi_x(z))$  - критерий качества аппроксимации  $\hat{F}$  функцией  $g$  в окрестности  $x$  определяемой  $\Pi_x(z)$

$\Omega(g)$  - мера сложности интерпретации функции  $g$ .

LIME:  $\xi(x) = \arg \min_{g \in G} (\mathcal{L}(\hat{F}, g, \Pi_x(z)) + \Omega(g))$

# LIME для текстов

---

В качестве  $\Pi_x(z)$  используем любое расстояние между текстами, а в качестве интерпретируемой репрезентации текста - bag of words.

# LIME для текстов

---

В качестве  $\Pi_x(z)$  используем любое расстояние между текстами, а в качестве интерпретируемой репрезентации текста - bag of words.

1. Генерируем выборку размера  $N$  в окрестности  $x$  в терминах  $\Pi_x(z)$  (можно, например, заменять несколько токенов исходного предложения на случайные).

# LIME для текстов

---

В качестве  $P_x(z)$  используем любое расстояние между текстами, а в качестве интерпретируемой репрезентации текста - bag of words.

1. Генерируем выборку размера  $N$  в окрестности  $x$  в терминах  $P_x(z)$  (можно, например, заменять несколько токенов исходного предложения на случайные).
2. Получаем  $N$  таргетов с помощью базового алгоритма на объектах из сгенерированной выборки.



# LIME для текстов

---

В качестве  $P_x(z)$  используем любое расстояние между текстами, а в качестве интерпретируемой репрезентации текста - bag of words.

1. Генерируем выборку размера  $N$  в окрестности  $x$  в терминах  $P_x(z)$  (можно, например, заменять несколько токенов исходного предложения на случайные).
2. Получаем  $N$  таргетов с помощью базового алгоритма на объектах из сгенерированной выборки.
3. Отбираем  $K$  самых важных слов в bag of words с помощью коэффициентов Lasso-регрессии, обученной на сгенерированном датасете.

# LIME для текстов

---

В качестве  $P_x(z)$  используем любое расстояние между текстами, а в качестве интерпретируемой репрезентации текста - bag of words.

1. Генерируем выборку размера  $N$  в окрестности  $x$  в терминах  $P_x(z)$  (можно, например, заменять несколько токенов исходного предложения на случайные).
2. Получаем  $N$  таргетов с помощью базового алгоритма на объектах из сгенерированной выборки.
3. Отбираем  $K$  самых важных слов в bag of words с помощью коэффициентов Lasso-регрессии, обученной на сгенерированном датасете.
4. Обучаем любую интерпретируемую модель на  $K$  отобранных признаках (обычно линейную регрессию с MSE).

## Пример работы LIME

### Example #3 of 6

True Class: ● Atheism

Instructions

[Previous](#)

Next

### Algorithm 1

**Words that A1 considers important:**

A horizontal bar chart with the following data:

Word	Frequency (approximate)
GOD	10
mean	8
anyone	7
this	6
Koresh	4
through	3

**Predicted:**

● Atheism

**Prediction correct:**



### Algorithm 2

**Words that A2 considers important:**

Part of Speech	Count (approx.)
Posting	100
Host	95
Re	45
by	40
in	40
Nntp	25

**Predicted:**

● Atheism

**Prediction correct:**



## Document

From: pauld@verdix.com (Paul Durbin)  
Subject: Re: DAVID CORESH IS! **GOD!**  
Nntp-Posting-Host: sarge.hq.verdix.com  
Organization: Verdix Corp  
Lines: 8

## Document

From: pauld@verdix.com (Paul Durbin)  
Subject: **Re: DAVID CORESH IS! GOD!**  
**Nntp-Posting-Host:** sarge.hq.verdix.com  
Organization: Verdix Corp  
Lines: 8

# Достоинства и недостатки LIME

---

## Плюсы:

- + Не зависит от внутреннего устройства модели
- + Легок в реализации
- + Human-friendly результаты

## Минусы:

- Не всегда легко найти подходящее  $P_x(z)$
- Влияние случайности на интерпретацию
- Не всегда стабилен

# SHapley Additive exPlanation values (SHAP values)

---

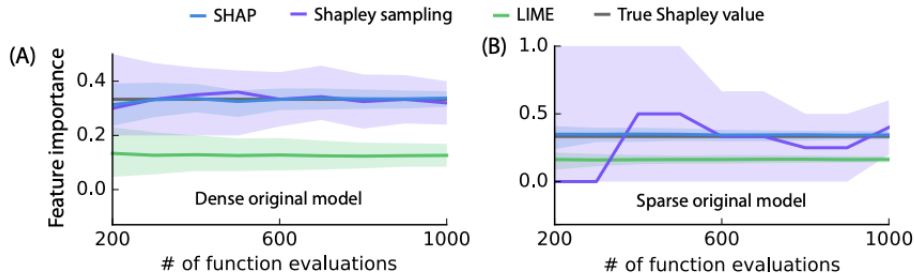
**Идея:**

Попробуем посчитать Shapley value для каждого признака.

**Формально:**

Shapley value: 
$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

# Сравнение различных методов расчета Shapley value



# White-box методы

# Gradient-based подход к интерпретации (Attribution)

---

**Идея:**

Давайте считать важность каждого входного токена как нормированный градиент функции потерь по этому токenu.

**Формально:**

$$a_i(x) = \frac{|\Delta_{x_i} \mathcal{L}(x) x_i|}{\sum_{j=1}^d |\Delta_{x_j} \mathcal{L}(x) x_j|}$$



# Gradient-based подход к интерпретации (Integrated Grads)

---

## Идея:

Хотим метод, в котором соблюдается несколько утверждений:

1. Если есть два объекта, которые отличаются только в одном признаке, но модель выдает на них разный ответ, то важность этого признака не должна быть нулевой.
2. Если есть две модели с разным внутренним устройством, но одинаковыми выходом на всех входах, то интерпретироваться они должны одинаково.

# Gradient-based подход к интерпретации (Integrated Grads)

---

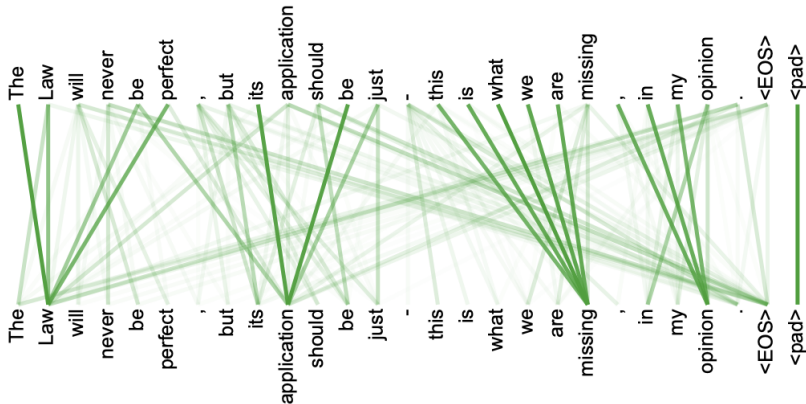
Формально:

$f$  - модель,  $x'$  - бейзлайн объект (пустая строка)

$$\text{IntegratedGrads}_i(x) = (x - x')_i \int_0^1 \frac{\partial f(x' - \alpha(x - x'))}{\partial x_i} d\alpha$$

$$\text{IntegratedGrads}_i^{\text{approx}}(x) = \frac{1}{M+1} \sum_{i=0}^M \frac{\partial f(x' - \frac{i}{M}(x - x'))}{\partial x_i}$$

# Визуализация attention слоев



# Как оценить качество интерпретации?

---

- **Оценка экспертом**

Например, качество интерпретации классификатора болезней может оценить врач-диагност.

- **Человеческая оценка**

Оцениваем качество интерпретации с помощью простых экспериментов с участием человека. Например, можно просить людей выбирать из двух разных интерпретаций лучшую.






- **Функциональная оценка**

Не требует участия человека. Используются различные метрики, в зависимости от задачи. Основная сложность именно в том, чтобы определить хорошую метрику качества.

# Готовые решения для интерпретации


# Список источников

---

-  Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. Ann. Statist. 29 (5) 1189 - 1232, October 2001
-  Alex Goldstein, Adam Kapelner, Justin Bleich, Emil Pitkin. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. 2013
-  Aaron Fisher, Cynthia Rudin, Francesca Dominici. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. 2018
-  Scikit learn documentation. 4.2. Permutation feature importance.
-  Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. 2016

# Список источников

---

-  Mukund Sundararajan, Ankur Taly, Qiqi Yan. Axiomatic Attribution for Deep Networks. 2017
-  Joseph D. Janizek, Pascal Sturmfels, Su-In Lee. Explaining Explanations: Axiomatic Feature Interactions for Deep Networks. 2020
-  Transformers interpret
-  Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, Ann Yuan. The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models. 2020
-  Alammur, J. Interfaces for Explaining Transformer Language Models [Blog post]. 2020
-  Xiang Zhou. Interpretability Methods in Machine Learning: A Brief Survey

# Список источников

---



Junlin Wang, Jens Tuyls, Eric Wallace, Sameer Singh. Gradient-based Analysis of NLP Models is Manipulable. 2020