Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики

Работа допущена к защите

Директор ВШПМиВФ ИПММ

_____Л.В. Уткин

17 июня 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

РАБОТА БАКАЛАВРА

РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМА ИНТЕРПРЕТАЦИИ РЕЗУЛЬТАТОВ ФУНКЦИОНИРОВАНИЯ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ МОДЕЛИ ЗАСОРЕНИЯ

по направлению подготовки 02.03.01 – Математика и компьютерные науки

Направленность (профиль) 02.03.01_01 – Вычислительные, программные, информационные системы и компьютерные технологии

Выполнил
студент гр. 3630201/70101 К.А. Вишняков

Руководитель
Директор ВШПМиВФ ИПММ,
д.т.н., профессор Л.В. Уткин

Консультант
по нормоконтролю И.Э. Голубева

Санкт-Петербург – 2021

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Институт прикладной математики и механики

УТВЕРЖДАЮ

Директор ВШПМиВФ ИПММ

_____Л.В. Уткин

1 марта 2021 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Вишнякову Кириллу Александровичу группы 3630201/70101

1. Тема работы: Разработка и исследование алгоритма интерпретации результатов функционирования моделей машинного обучения с использованием модели засорения.
2. Срок сдачи студентом законченной работы: июнь 2021.
3. Исходные данные по работе:
   3.1. Наборы реальных данных для проведения экспериментов.
   3.2. Постановка оптимизационной задачи для вычисления нижних и верхних границ чисел Шепли.
   3.3. Постановка оптимизационной задачи для сужения интервалов.
   3.4. Функция для вычисления расстояний между распределениями вероятностей.
4. Содержание работы (перечень подлежащих разработке вопросов):
   4.1. Изучение метода интерпретируемого машинного обучения SHAP.
   4.2. Разработка и реализация алгоритма.
   4.3. Проведение экспериментов на синтетических и реальных данных.
   4.4. Анализ полученных результатов.
5. Перечень графического материала (с указанием обязательных чертежей):
   5.1. Графики экспериментов, включающие в себя тепловые карты и диаграммы.

6. Консультанты по работе: -.

7. Дата выдачи задания 1 марта 2021 г.

Руководитель ВКР Л.В. Уткин

Задание принял к исполнению 1 марта 2021 г.

Студент К.А. Вишняков

РЕФЕРАТ

На 39 с., 30 рисунков, 4 таблицы.

КЛЮЧЕВЫЕ СЛОВА: МАШИННОЕ ОБУЧЕНИЕ, ИСКУСТВЕННЫЙ ИНТЕЛЛЕКТ, ОБЪЯСНИТЕЛЬНЫЙ ИСКУСТВЕННЫЙ ИНТЕЛЛЕКТ, МОДЕЛЬ ЗАСОРЕНИЯ, SHAP.

Объяснение предсказаний моделей машинного обучения является одной из важнейших задач в таких областях применения искусственного интеллекта как медицина, финансы и управление автономными системами. Одним из основных методов объяснительного искусственного интеллекта является метод SHAP. Обычная версия SHAP не предназначена для работы с распределениями вероятностей, что затрудняет интерпретацию моделей, решающих задачу многоклассовой классификации. Еще одним недостатком стандартного SHAP является то, что он не учитывает неточность модели при вычислении значений чисел Шепли. В работе описан способ реализации метода SHAP на случай многоклассовой классификации, в котором рассматриваются множества распределений вероятностей, построенных в соответствии со статистической моделью ϵ-засорения, что позволяет получить интервалы значений чисел Шепли вместо одиночных значений. По результатам экспериментов, было показано, что одиночные значения чисел Шепли из-за возможной неточности предсказаний модели и возникающей неопределенности при их вычислении, часто выдают ненадежные результаты, и вместо них должны рассматриваться интервалы значений.

ABSTRACT

39 pages, 30 pictures, 4 tables.

KEYWORDS: MACHINE LEARNING, ARTIFICIAL INTELLIGENCE, EXPLAINABLE ARTIFICIAL INTELLIGENCE, CONTAMINATION MODEL, SHAP.

Explanation of the predictions of machine learning models is one of the most important tasks in such areas of artificial intelligence applications as medicine, finance, and control of autonomous systems. One of the main methods of explainable artificial intelligence is the SHAP method. The regular version of the SHAP is not intended to work with probability distributions, which makes it difficult to interpret models that solve the problem of multiclass classification. Another disadvantage of the standard SHAP is that it does not take into an account the imprecision of the model's predictions when calculating the values of the Shapley numbers. This work describes a method for implementing the SHAP algorithm for the case of multiclass classification problem, which considers sets of distributions constructed in accordance with the statistical ϵ-contamination model, which allows obtaining intervals of Shapley numbers instead of single values. The experiments are presented showing that single values of the Shapley numbers often give unreliable results, due to the possible inaccuracy of the model's predictions and the resulting uncertainty in their calculation, and instead of them, intervals of values should be considered.

СОДЕРЖАНИЕ

Введение ... 8

Глава 1. Методы интерпретируемого машинного обучения ... 10

1.1. Классификация методов интерпретируемого машинного обучения ... 10

1.2. Метод внешней интерпретации SHAP ... 11

Глава 2. Интервальный SHAP ... 14

2.1. Модель ϵ-засорения ... 14

2.2. Задача классификации ... 14

2.3. Интервалы значений чисел Шепли ... 14

2.4. Сужение интервалов ... 19

Глава 3. Реализация ... 20

Глава 4. Эксперименты ... 21

4.1. Модель ... 21

4.2. Описание датасетов ... 21

4.2.1. Окружность с двумя классами ... 21

4.2.2. Окружность с тремя классами ... 22

4.2.3. Четыре кластера данных ... 23

4.3. Описание экспериментов ... 24

4.3.1. Датасет Окружность с двумя классами ... 24

4.3.2. Датасет Окружность с тремя классами ... 31

4.3.3. Датасет Четыре кластера данных ... 35

4.3.4. Датасет Seeds ... 38

4.3.5. Датасет Ecoli ... 41

4.3.6. Датасет Glass Identification 42

Заключение 45

Список сокращений и условных обозначений..... 47

Список использованных источников 48

ВВЕДЕНИЕ

В последние был достигнут большой прогресс в области машинного обучения и искусственного интеллекта. Во многом это стало возможным благодаря разработке новых более сложных и громоздких моделей, таких как, например, глубокие нейронные сети. Несмотря на то, что такие модели зачастую демонстрируют высокую точность своих предсказаний, у них есть один существенный недостаток. Этот недостаток заключается в том, что со стороны пользователя эти модели выглядят как черный ящик, то есть пользователь может лишь дать данные на вход и получить для них предсказание. Тем самым, модель никак не объясняет почему она вынесла то или иное предсказание. В то же время, в некоторых областях применения моделей машинного обучения, объяснение предсказания может быть не менее важным, чем само предсказание. К примеру, в медицинских приложениях [1][16][6], модель, которая имеет высокую точность и хорошо объяснима, не только сможет диагностировать то или иное заболевание, но также поможет врачу выбрать более эффективные методы лечения. Объяснять предсказание модели машинного обучения также требуется в сфере финансов [3][2] и при эксплуатировании автономных систем [7][4].

В связи с этим в последние годы очень актуальным стало направление интерпретируемого машинного обучения. В этом контексте интерпретируемость [9] – это величина, которая говорит, насколько легко человек может понять причину, по которой было вынесено то или иное предсказание. Как следствие, было разработано множество методов интерпретируемого машинного обучения. Одним из основных методов интерпретируемого машинного обучения, который относится к классу локальных, является метод SHAP [8].

Одной из проблем метода SHAP является то, что он не работает для моделей, которые решают задачу многоклассовой классификации. Это является достаточно серьезным ограничением, так как задача многоклассовой

классификации встречается в машинном обучении повсеместно. Также, общей проблемой всех методов интерпретируемого машинного обучения является то, что сами модели зачастую могут давать не очень точные предсказания, что в свою очередь, приводит к тому, что методы интерпретируемого машинного обучения также будут давать неточные результаты. Стоит отметить, что в методе SHAP неопределенность также добавляется, в момент вычисления $f(S)$. Это происходит из-за того, что значения опущенных признаков заменяются какими-то константными значениями, которые может задать пользователь. Поэтому, чтобы учесть неточность и неопределенность предсказания модели вместо обычного распределения вероятностей, которое модель дает на выходе, в данной работе рассматриваются множества возможных распределений вероятностей. Такие множества распределений вероятностей могут быть построены в соответствии с одной из статистических моделей засорения. Одной из таких моделей является модель ϵ-засорения.

В данной работе рассматривается как можно реализовать метод SHAP на случай многоклассовой классификации и учесть неточности предсказания модели, используя модель ϵ-засорения.

Целью данной работы является: разработка алгоритма, на основе SHAP для задачи многоклассовой классификации, позволяющего учесть неопределенность предсказаний модели, используя модель ϵ-засорения.

Задачами данной работы является:

- исследование методов интерпретируемого машинного обучения;
- разработка и реализация расширения алгоритма SHAP для случая многоклассовой классификации с использованием модели ϵ-засорения;
- генерация и подготовка наборов данных, необходимых для проведения экспериментов. Проведение экспериментов и тестирование разработанного алгоритма на синтетических и реальных наборах данных;
- анализ результатов проведенных экспериментов.

ГЛАВА 1. МЕТОДЫ ИНТЕРПРЕТИРУЕМОГО МАШИННОГО ОБУЧЕНИЯ

1.1. Классификация методов интерпретируемого машинного обучения

На данный момент существует большое множество различных моделей в машинном обучении: линейные модели, деревья решений, случайные леса, SVM, глубокие нейронные сети. С точки зрения интерпретируемого машинного обучения эти модели можно разделить на два класса: самоинтерпретируемые модели и модели для интерпретации которых требуется использовать дополнительные внешние методы.

Линейные модели и простые деревья решений зачастую рассматриваются как самоинтерпретируемые модели. К примеру, для линейных моделей достаточно посмотреть на ее коэффициенты, которые стоят перед переменными, чтобы понять, как каждый признак влияет на результирующее предсказание. Для простейших деревьев решений можно изучить условия, которые встречаются на пути от корня листа. Однако часто бывают так, что даже такие модели не очень поддаются интерпретации. Обычно это происходит, когда входные данные являются векторами из разреженных многомерных пространств и поэтому для таких моделей требуется разрабатывать специальные методы [10]. Например, в случае высокой размерности входных данных для линейных моделей большое количество коррелируемых величин и закономерностей представляются одним и тем же конечным набором коэффициентов, что в свою очередь снижает интерпретируемость каждого коэффициента в отдельности. В то же время, для того чтобы получить хорошую точность для деревьев решений при работе с данными большой размерности требуется строить деревья большой глубины, что также ведет к увеличению числа узлов и случаям, когда на одном пути от корня до листа один и тот же признак встречается в нескольких условиях, что делает объяснение для конечного пользователя более запутанным.

К другому классу моделей для которых требуется использование внешних методов интерпретации, относятся глубокие нейронные сети, SVM и случайные леса. Из-за своей внутренней сложности эти модели со стороны пользователя выглядят как черный ящик, поэтому зачастую для их интерпретации требуется использование внешних методов.

Сами методы интерпретируемого машинного обучения могут быть классифицированы на методы, которые предназначены для объяснения моделей машинного обучения конкретного типа, к примеру Interpretable Neural Embeddings [15], DeepLift [14], Grad-CAM [12], и методы, которые работают для широкого спектра моделей, такие как LIME[11], SHAP и Permutation Feature Importance[5].

Методы интерпретируемого машинного обучения можно также разделить на две группы в зависимости от того какое объяснение они дают: глобальное или локальное. Например, Permutation Feature Importance является глобальным методом, так как объясняет поведение модели в целом, а SHAP и LIME являются локальными методами, так как объясняют предсказание модели для одного конкретного примера.

1.2. Метод внешней интерпретации SHAP

Метод SHAP основан на вычислении чисел Шепли [13] - концепта из теории коалиционных игр. Числа Шепли позволяют определить как справедливо распределить прибыль и затраты между участниками, когда участники работают в коалиции.

В контексте машинного обучения в роли выигрыша рассматривается предсказание модели для заданного примера, а в роли игроков выступают признаки. Таким образом, вычислив числа Шепли можно оценить вклад каждого из признаков в результирующее предсказание модели. Вклад i-го признака в предсказание модели f определяется по следующей формуле:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} B(S, N)[f(S \cup \{i\}) - f(S)], \qquad (1.1)$$

где $f(S)$ – это предсказание модели черного ящика на подмножестве признаков S;

N – множество всех признаков;

$B(S, N)$ задается как:

$$B(S,N) = \frac{|S|!\,(|N| - |S| - 1)!}{|N|}.$$

Число Шепли ϕ_i можно понимать, как средний вклад i-го признака в финальное предсказание, учтенный по всем возможным комбинациям, где участвовал i-ый признак.

Числа Шепли обладают следующими свойствами:

1. Эффективность. То есть, что сумма всех вкладов равна разности предсказания модели $f(x)$ и $f(\emptyset)$:

$$\sum_{k=0}^{m} \phi_k = f(x) - f(\emptyset). \quad (1.2)$$

2. Симметричность. Если два игрока с номерами i и j внесли одинаковый вклад, то есть для всех подмножеств S, которые не включают i и j, выполняется $f(S \cup \{i\}) = f(S \cup \{j\})$, то верно, что $\phi_i = \phi_j$.
3. Аксиома нулевого игрока. Если игрок под номером j, для любого подмножества признаков S, вносит нулевой вклад, то есть, $f(S \cup \{j\}) = f(S)\ \forall S \subseteq N \setminus \{j\}$, тогда верно, что $\phi_j = 0$.
4. Линейность. Если некоторая игра f представляется как линейная комбинация нескольких игр $f_1, \dots, f_n$, то $\forall i$ верно, что

$$\phi_i(f) = \sum_{k=1}^{m} c_k \phi_i(f_k).$$

Так как количество слагаемых в сумме для вычисления ϕ_i достаточно велико, то были предложены методы, которые аппроксимируют значение ϕ_i и тем самым уменьшают время работы алгоритма, например Max SHAP и Deep SHAP [4].

В методе SHAP так же имеет значение каким именно образом, будут заменены удаленные признаки, то есть признаки, не входящие во множество S, чтобы вычислить $f(S)$. Для табличных данных значения отсутствующих признаков можно заменять нулями, средними значениями признаков, значениями, которые задает сам пользователь, или значениями, которые генерируются из какого-либо ранее заданного распределения.

Стоит отметить, что SHAP работает не только с табличными данными, но и с изображениями и текстом. Для изображений все пиксели группируются в непрерывные наборы так называемых суперпикселей, которые и выступают в качестве признаков. В таком случае недостающие суперпиксели можно генерировать с помощью дополнительной генеративной модели. Для текстовых данных в качестве признаков выступают лексемы.

ГЛАВА 2. ИНТЕРВАЛЬНЫЙ SHAP

2.1. Модель ϵ-засорения

Для того, чтобы учесть неточность предсказаний интерпретируемой модели нужно рассматривать не одно единственное распределение вероятностей, которое является предсказанием модели, а некоторое множество таких распределений. Одним из способов задания таких распределений является модель ϵ-засорения.

Модель ϵ-засорения генерирует множество распределений вероятностей $P'(\epsilon, P)$, где каждое распределение имеет вид $P^* = (p_1^*, \ldots, p_C^*)$. Каждая вероятность из p_i^* определяется формулой:

$$p_i^* = (1 - \epsilon)p_i + \epsilon h_i,$$

где $P = (p_1, \ldots, p_C)$ является распределением вероятностей, которое получено для примера для которого требуется объяснить предсказание модели;

h - единичный симплекс;

ϵ - параметр засорения.

Для этой модели выполняются условия, что $h_i \geq 0, h_1 + \cdots + h_C = 1$ и $0 \leq \epsilon \leq 1$. Параметр ϵ контролирует размер множества P' и может быть определен из размера выборки для обучения. Чем больше примеров в обучающей выборке, тем точнее определены значения распределения вероятностей и тем меньше должно быть значение параметра ϵ. В построенном множестве P' каждое распределение, которое принадлежит этому множеству, является кандидатом на то, чтобы быть неким истинным распределением предсказания модели, которое на самом деле является неизвестным.

2.2. Задача классификации

Так как оригинальный метод SHAP работает не с распределениями вероятностей, а с одной вероятностью, то он не применим для задач многоклассовой классификации. Проблема заключается в том, что неясно как

вычислить разность $f(S \cup i) - f(S)$, когда $f(X)$ является не действительным числом, а распределением вероятностей.

Одной из метрик для сравнения распределений вероятностей является расстояние Кульбака-Лейблера, которое для дискретного случая распределения вероятностей определяется по следующей формуле:

$$D_{KL}(P, Q) = \sum_{i=1}^{n} p_i \cdot log \frac{p_i}{q_i},$$

где P и Q это два распределения с вероятностями p_i и q_i соответственно. Чем меньше величина этого расстояния, тем более похожи два распределения и наоборот.

Другой метрикой, которая используется для сравнения распределений вероятностей является метрика Колмогорова-Смирнова, она задается следующим образом:

$$D_{KS}(P, Q) = \max_{i=1,\dots,C-1} |\pi_i - \tau_i|,$$

где $\pi_i = \sum_{j=1}^{i} p_j$ и $\tau_i = \sum_{j=1}^{i} q_j$. Таким образом, формулу (1.1) можно переписать в следующем виде:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} B(S, N) D\big(P(S \cup \{i\}), P(S)\big),$$

где в роли D может выступать расстояние Кульбака-Лейблера, Колмогорова-Смирнова или другая функция, предназначенная для сравнения распределений вероятностей.

Для того, чтобы сохранилось свойство эффективности чисел Шепли, нельзя просто считать расстояние между распределениями $P(S \cup \{i\})$ и $P(S)$. Поэтому вводиться опорная точка, до которой попарно считаются расстояния от $P(S \cup \{i\})$ и $P(S)$. В качестве опорной точки выступает сам объясняемый пример. Из-за введения опорной точки требуется поменять знаки в разности при вычислении чисел Шепли. Таким образом для случая распределения вероятностей числа Шепли будут вычисляться по следующей формуле:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} B(S,N)\left[D(P_S, P_N) - D\left(P_{S \cup \{i\}}, P_N\right)\right]. \quad (2.1)$$

Тогда свойство эффективности для случая распределения вероятностей действительно будет выполняться:

$$\sum_{k=0}^{m} \phi_k = f(x) - f(\emptyset) = D(P_\emptyset, P_N) - D(P_N, P_N) = D(P_\emptyset, P_N).$$

2.3. Интервалы значений чисел Шепли

Из-за использования модели ϵ-засорения изменяется способ определения расстояния между распределениями вероятностей. Поскольку существует несколько множеств распределений, то все возможные расстояния между парами элементов этих множеств создают интервал возможных значений чисел Шепли ϕ_i, обозначаемый как $\left[\phi_i^L, \phi_i^U\right]$. Таким образом формула (2.1) может быть переписана следующим образом:

$$\phi_i^L = \sum_{S \subseteq N \setminus \{i\}} B(S,N) \min_{P \in P'(P_S),\ R \in P'(P_S \cup \{i\}),\ Q \in P'(P_N)} \left(D(P,Q) - D(R,Q)\right)$$

$$\phi_i^U = \sum_{S \subseteq N \setminus \{i\}} B(S,N) \max_{P \in P'(P_S),\ R \in P'(P_S \cup \{i\}),\ Q \in P'(P_N)} \left(D(P,Q) - D(R,Q)\right),$$

где P, Q, R являются вероятностными распределениями, взятыми из подмножеств $P'(P_S), P'\left(P_{S \cup \{i\}}\right), P'(P_N)$.

Важно заметить, что если взять минимум и максимум раздельно, то есть:

$$\phi_i^L = \sum_{S \subseteq N \setminus \{i\}} B(S,N) \min_{P \in P'(P_S),\ Q \in P'(P_N)} D(P,Q) - \max_{R \in P'(P_S \cup \{i\}),\ Q \in P'(P_N)} D(R,Q)$$

$$\phi_i^U = \sum_{S \subseteq N \setminus \{i\}} B(S,N) \max_{P \in P'(P_S),\ Q \in P'(P_N)} D(P,Q) - \min_{R \in P'(P_S \cup \{i\}),\ Q \in P'(P_N)} D(R,Q),$$

то полученные интервалы значений чисел Шепли получатся очень широкими, так как будут рассмотрены случаи, когда распределение $Q \in P'(P_N)$ будет определено по-разному для расстояний $D(P,Q)$ и $D(R,Q)$. Поэтому, предполагается, что есть единственное распределение вероятностей классов Q, для которого известно, что оно лежит в $P'(P_N)$ и для него достигается

наименьшее и наибольшее значение $D(P,Q) - D(R,Q)$. Таким образом, могут быть вычислены границы интервалов для значений чисел Шепли. Далее, свойство эффективности чисел Шепли может быть использовано для сужения этих интервалов. В качестве D будет выступать расстояние Колмогорова-Смирнова, так как она позволяет свести решение оптимизационных задач к решению простого набора задач линейного программирования.

Обозначим множества кумулятивных распределений π, τ, α, полученных для вероятностных распределений $P_S, P_{S\cup\{i\}}, P_N$, как $P'(\epsilon, P_S), P'\left(\epsilon, P_{S\cup\{i\}}\right), P'(\epsilon, P_N)$. Тогда будут выполняться следующие неравенства для нижних и верхних границ для $i = 1, \dots, C-1$:

$$\pi_i^L \le \pi \le \pi_i^U, \tau_i^L \le \tau \le \tau_i^U, \alpha_i^L \le \alpha \le \alpha_i^U \tag{2.2}$$

где $\pi_i^L, \tau_i^L, \alpha_i^L$ – нижние границы значений кумулятивных распределений, а $\pi_i^U, \tau_i^U, \alpha_i^U$ – верхние.

Заметим, что ограничение для $i = C$ не требуется, так как из того, что π, τ, α являются кумулятивными распределениями, всегда будет следовать, что $\pi_C = \tau_C = \alpha_C = 1$.

Рассмотрим нижнюю границу L для $D_{KS}(P,Q) - D_{KS}(R,Q)$. Ее значение можно получить, решив следующую оптимизационную задачу:

$$L = \min_{\pi,\tau,\alpha} \left(\max_{i=1,\dots,C-1} |\pi_i - \alpha_i| - \max_{i=1,\dots,C-1} |\tau_i - \alpha_i| \right), \tag{2.3}$$

учитывая условия, описанные в (2.2). Эта задача является задачей невыпуклой оптимизации, однако она может быть представлена множеством из $2C-2$ простых задач линейного программирования.

Нижнюю границу L для $D_{KS}(P,Q) - D_{KS}(R,Q)$ можно вычислить, решив следующую систему из $2C-2$ задач линейного программирования:

$$L_1(k) = \min_{B,\pi,\alpha} (B - \tau_k^U + \alpha_k), k = 1, \dots, C-1 \tag{2.4}$$

$$L_2(k) = \min_{B,\pi,\alpha} (B - \alpha_k + \tau_k^L), k = 1, \dots, C-1 \tag{2.5}$$

где $B = \max_{i=1,\dots,C-1} |\pi_i - \alpha_i|$, относительно общих условий:

$$\begin{cases} \pi_i^L \le \pi_i \le \pi_i^U, \alpha_i^L \le \alpha_i \le \alpha_i^U, i = 1, \dots, C-1 \\ B \ge \pi_i - \alpha_i, B \ge \alpha_i - \pi_i, i = 1, \dots, C-1 \\ \pi_i \le \pi_{i+1}, \alpha_i \le \alpha_{i+1}, i = 1, \dots, C-2, \end{cases}$$

также для задач типа L_1 будут добавлены следующие условия:

$$\begin{cases} \alpha_k \le \tau_k^U \\ \tau_k^U - \alpha_k \ge \tau_i^U - \alpha_i, i = 1, \dots, C-1, i \ne k, \end{cases}$$

а для задач типа L_2 будут добавлены следующие условия:

$$\begin{cases} \alpha_k \ge \tau_k^L \\ \alpha_k - \tau_k^L \ge \alpha_i - \tau_i^L, i = 1, \dots, C-1, i \ne k. \end{cases}$$

Далее, нужно сначала выбрать $k_1 = argmax_k(\tau_k^U - \alpha_k)$ из (2.4) и $k_2 = argmax_k(\alpha_k - \tau_k^L)$ из (2.5), тогда окончательное решение L определяется как $L = L_1(k_1)$, если $\tau_{k_1}^U - \alpha_{k_1} \ge \alpha_{k_2} - \tau_{k_2}^L$, иначе $L = L_2(k_2)$.

Рассмотрим нижнюю границу U для $D_{KS}(P,Q) - D_{KS}(R,Q)$. Ее значение можно получить, решив следующую оптимизационную задачу:

$$U = \max_{\pi,\tau,\alpha} \left(\max_{i=1,\dots,C-1} |\pi_i - \alpha_i| - \max_{i=1,\dots,C-1} |\tau_i - \alpha_i| \right), \quad (2.6)$$

учитывая условия, описанные в (2.2).

Верхняя граница U для $D_{KS}(P,Q) - D_{KS}(R,Q)$ можно вычислить, решив следующую систему из 2C − 2 задач линейного программирования.

$$U_1(k) = \min_{B,\pi,\alpha} (\pi_k^U - \alpha_k - B), k = 1, \dots, C-1 \quad (2.7)$$

$$U_2(k) = \min_{B,\pi,\alpha} (\alpha_k - \pi_k^L + B), k = 1, \dots, C-1, \quad (2.8)$$

где $B = \max_{i=1,\dots,C-1} |\tau_i - \alpha_i|$, относительно общих условий:

$$\begin{cases} \tau_i^L \le \tau_i \le \tau_i^U, \alpha_i^L \le \alpha_i \le \alpha_i^U, i = 1, \dots, C-1 \\ B \ge \tau_i - \alpha_i, B \ge \alpha_i - \tau_i, i = 1, \dots, C-1 \\ \tau_i \le \tau_{i+1}, \alpha_i \le \alpha_{i+1}, i = 1, \dots, C-2, \end{cases}$$

Также для задач типа U_1 будут добавлены следующие условия:

$$\begin{cases} \alpha_k \le \pi_k^U \\ \pi_k^U - \alpha_k \ge \pi_i^U - \alpha_i, i = 1, \dots, C-1, i \ne k, \end{cases}$$

а для задач типа U_2 будут добавлены следующие условия:

$$\begin{cases} \alpha_k \geq \pi_k^L \\ \alpha_k - \pi_k^L \geq \alpha_i - \pi_i^L, i = 1, \dots, C-1, i \neq k. \end{cases}$$

Далее, нужно сначала выбрать $k_1 = argmax_k(\pi_k^U - \alpha_k)$ из (2.7) и $k_2 = argmax_k(\alpha_k - \pi_k^L)$ из (2.8), тогда окончательное решение U определяется как $U = U_1(k_1)$, если $\pi_{k_1}^U - \alpha_{k_1} \geq \alpha_{k_2} - \pi_{k_2}^L$, иначе $U = U_2(k_2)$.

2.4. Сужение интервалов

Полученные интервалы значений чисел Шепли можно сузить, использовав свойство эффективности. Формулу (2.2) для свойства эффективности можно переписать следующим образом, когда вычисляется расстояние между распределениями вероятностей:

$$\sum_{k=0}^{m} \phi_k = D(P_\emptyset, P_N).$$

Нижняя и верхняя границы для $D(P_\emptyset, P_N)$ будут обозначаться как D^L и D^U соответственно. Значения этих границ можно будет вычислить, решив следующие оптимизационные задачи:

$$D^L = \min_{P \in P'(P_\emptyset),\ Q \in P'(P_N)} D(P, Q) \tag{2.9}$$

$$D^U = \max_{P \in P'(P_\emptyset),\ Q \in P'(P_N)} D(P, Q). \tag{2.10}$$

Тогда свойство эффективности чисел Шепли можно будет переписать следующим образом:

$$D^L \leq \sum_{k=0}^{m} \phi_k \leq D^U.$$

Пусть $\sum_{k=0}^{m} \phi_k^U \geq D^U$ и $\sum_{k=0}^{m} \phi_k^{\mathrm{L}} \leq D^{\mathrm{L}}$, тогда границы суженных интервалов можно будет получить по следующим формулам:

$$\tilde{\phi}_k^U = min\left(\phi_k^U, D^U - \sum_{i=1, i \neq k}^{m} \phi_i^L\right) \tag{2.11}$$

$$\tilde{\phi}_k^L = min\left(\phi_k^L, D^L - \sum_{i=1, i \neq k}^{m} \phi_i^U\right). \tag{2.12}$$

ГЛАВА 3. РЕАЛИЗАЦИЯ

Весь код программы был написан на языке Python. Сначала был реализован стандартный алгоритм SHAP, работающий с одной вероятностью. Для реализации стандартного SHAP был разработан класс SimpleShap, имеющий конструктор, принимающий на вход модель, чьи предсказания нужно объяснить, и набор примеров из обучающей выборки. Этот класс имеет единственный метод calculate_shapley_values, который считает значения чисел Шепли для набора из тестовых примеров в формате pandas.DataFrame. После проверки, что результаты собственной реализация SHAP совпадают с результатами реализации, которая была разработана авторами самого метода SHAP, была начата разработка расширения стандартного алгоритма SHAP.

Для этого был разработан класс ImpreciseShap, имеющий такой же конструктор, как и SimpleShap. Различия между SimpleShap и ImpreciseShap заключаются в реализации метода calculate_shapley_values. В реализации этого метода в ImpreciseShap был добавлен код, который делает следующее:

- вычисление разности между $f(S \cup \{i\})$ и $f(S)$ с помощью метрики Колмогорова-Смирнова;
- решение оптимизационных задач (2.3) и (2.6);
- решение оптимизационных задач (2.9) и (2.10);
- вычисление суженных координат отрезков по формулам (2.11) и (2.12).

Для решения оптимизационных задач использовалась библиотека PuLP языка Python, которая использует библиотеку CBC, написанную на языке C++ в качестве бекенда. Для визуализации результатов и отрисовки графиков была использована библиотека Plotly.

ГЛАВА 4. ЭКСПЕРИМЕНТЫ

4.1. Модель

Во всех экспериментах в качестве модели был выбран классификатор Random Forest со следующими параметрам:

- n_estimators = 100;
- max_depth = 8;
- max_features = 2.

4.2. Описание датасетов

4.2.1. Окружность с двумя классами

- Тип: синтетический датасет.
- Размер датасета: 1000 примеров в обучающей выборке, 250 примеров в тестовой выборке.
- Количество признаков: 2.
- Распределение классов в обучающей выборке: 879 примеров - класс 0, 121 пример – класс 1;
- Распределение классов в тестовой выборке: 215 примеров - класс 0, 35 примеров - класса 1;
- Процесс генерации: датасет был сгенерирован с помощью семплирования из равномерного распределения для отрезка [0, 5]. Далее, точкам, лежащим внутри единичной окружности с центром (2.5, 2.5) был присвоен нулевой класс, а всем остальным точкам первый класс.

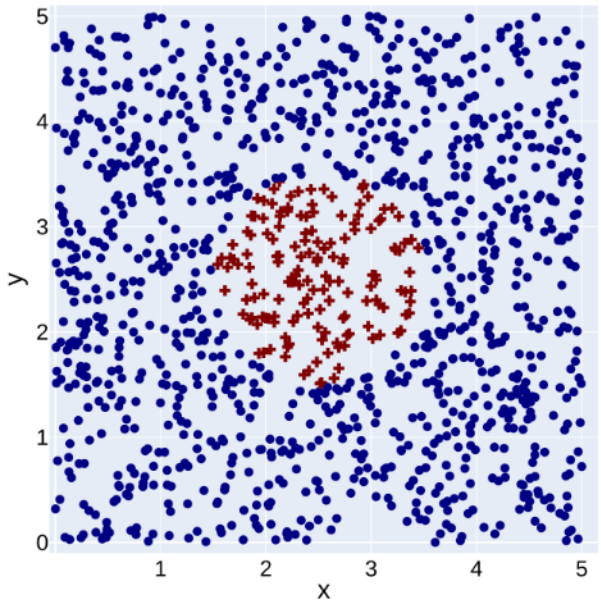

Рис. 4.1. Окружность с двумя классами

4.2.2. Окружность с тремя классами

- Тип: синтетический.
- Размер датасета: 1000 примеров в обучающей выборке, 250 примеров в тестовой выборке.
- Количество признаков: 2.
- Распределение классов в обучающей выборке: 340 примеров - класса 0, 336 примеров – класс 1, 324 примеров – класс 2.
- Распределение классов в тестовой выборке: 92 примера – класс 0, 80 примеров – класс 1, 78 примеров – класс 2.
- Процесс генерации: датасет был сгенерирован с помощью семплирования из нормального распределения с математическим ожиданием (2.5, 2.5) и матрицей ковариаций $0.5 \cdot I$, где I – это единичная матрица ковариаций. Далее, точкам, лежащим внутри единичной окружности с центром (2.5, 2.5), был присвоен нулевой класс. Точкам, лежащим вне окружности с центром в точке (2.5, 2.5) и радиусом 2 был присвоен второй класс, а всем остальным точкам был присвоен первый класс.

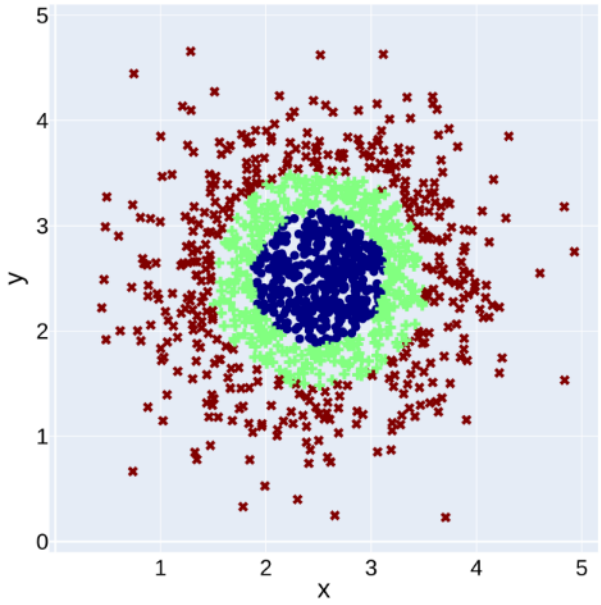

Рис. 4.2. Окружность с тремя классами

4.2.3. Четыре кластера данных

- Размер датасета: 1000 примеров в обучающей выборке, 250 примеров в тестовой выборке.
- Количество признаков: 2.
- Распределение классов в обучающей выборке: 249 примеров - класс 0, 263 примера – класс 1, 238 примеров – класс 2, 250 примеров – класс 3.
- Распределение классов в тестовой выборке: 64 примера – класс 0, 50 примеров – класс 1, 74 примера – класс 2, 62 примера - класс 3.
- Процесс генерации: кластеры данных расположены в вершинах квадрата со стороной 2. Внутри каждого из кластеров, координаты каждой точки получаются с помощью семплирования из нормального распределения.

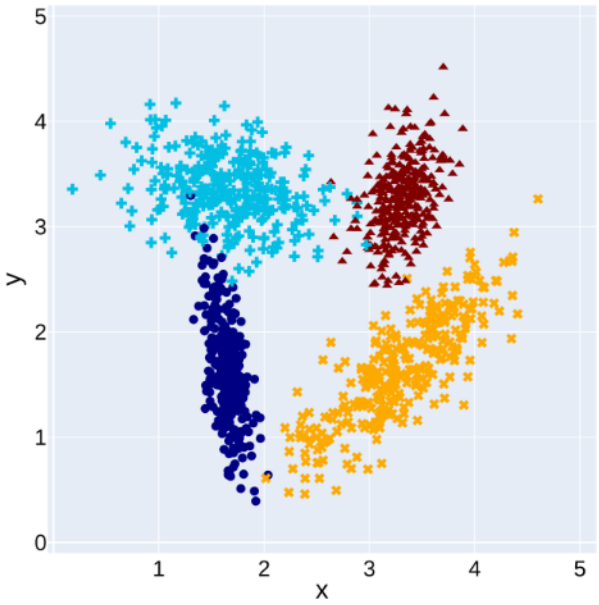

Рис. 4.3. Четыре кластера данных

4.3. Описание экспериментов

4.3.1. Датасет Окружность с двумя классами

Полученные значения чисел Шепли можно наглядно продемонстрировать на тепловой карте, где интенсивности света будет соответствовать значение числа Шепли для заданного признака. Так как для всех описанных выше датасетов, все координаты точек имеют значения в пределах от 0 до 5, то нужно разбить квадрат с вершинами в точках (0, 0), (0, 5), (5, 0), (5, 5) на одинаковые более мелкие квадраты одинаковой длины и ширины. Во всех последующих примерах, тепловая карта имеет размер 75x75 единиц, то есть сторона маленького квадрата равна 5 / 75 = 0.0666, а значение параметра ϵ, которое использовалось для построения тепловых карт равно 0.15. Далее, для каждого маленького квадрата, требуется взять точку, лежащую в его середине и посчитать для нее числа Шепли для каждого из признаков. Таким образом, будет построен набор тепловых карт, показывающий зависимость значений чисел Шепли от значений признаков (в случае двумерной плоскости значений x и y координат), где:

- $\phi(f_l)$ – значение левой границы числа Шепли для признака f;
- $\phi(f_u)$ – значение правой границы числа Шепли для признака f;
- $\big(\phi(f_l) + \phi(f_u)\big)/2$ – значение середины отрезка возможных значений числа Шепли для признака f;
- $len\ \phi(f)$ – значение длины отрезка.

Во всех тепловых картах для величин $\phi(f_l), \phi(f_u), \big(\phi(f_l) + \phi(f_u)\big)/2$, если в какой-либо точке их значение выходит за переделы отрезка $[-0.5, 0.5]$, то интенсивность света будет соответствовать интенсивности света, при значении -0.5 или 0.5 в зависимости от того в какую сторону произошло переполнение. Тот же принцип выполняется для величины $len\ \phi(f)$ и отрезка $[0, 0.5]$.

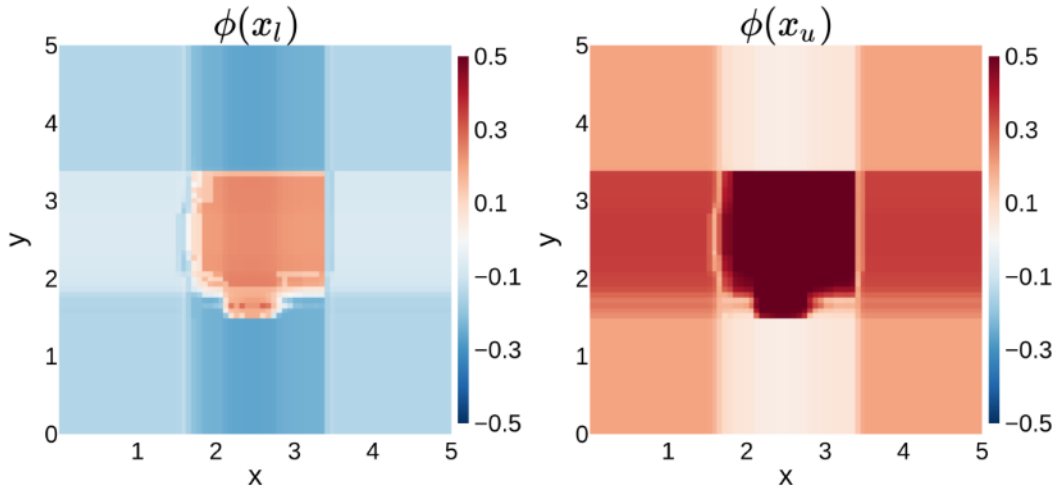

Рис. 4.4. Тепловые карты для значений левой и правой границы для признака x

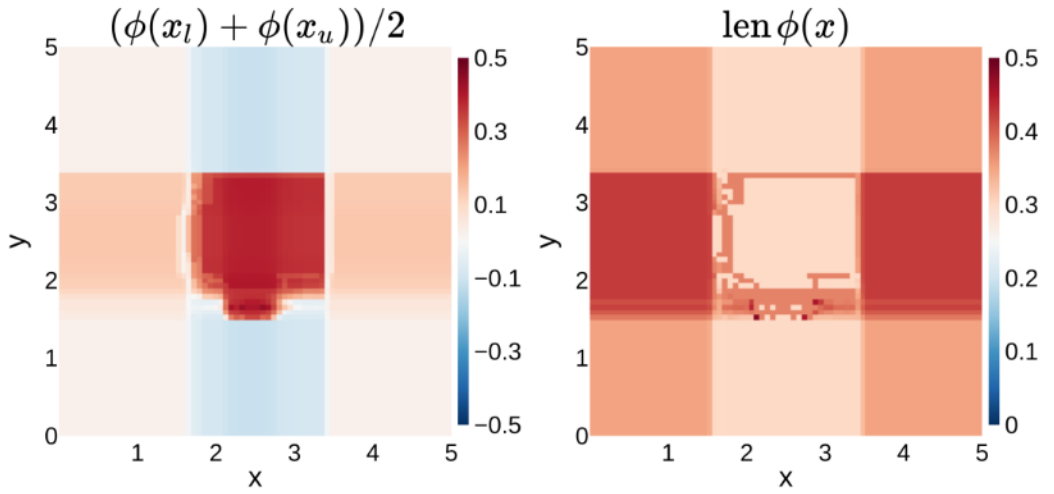

Рис. 4.5. Тепловые карты для значения середины и длины отрезка для признака x

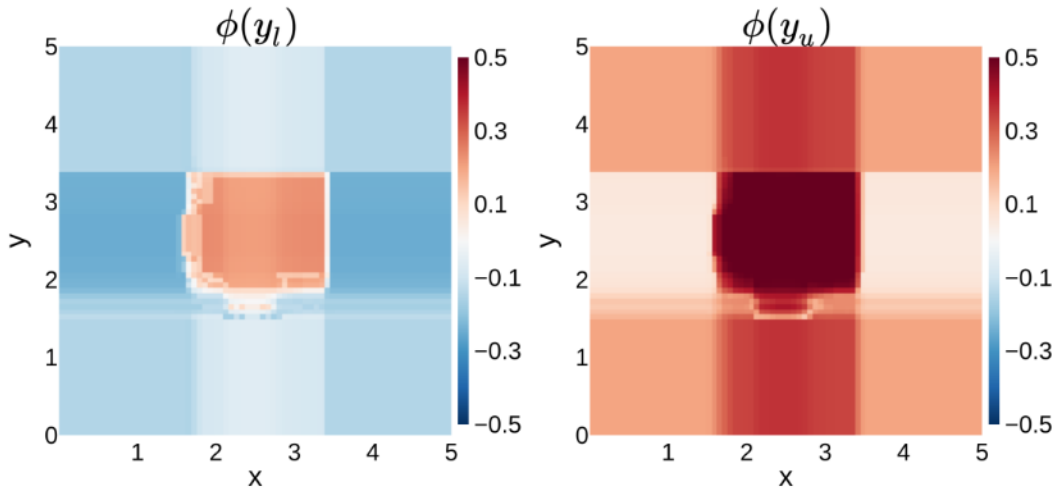

Рис. 4.6. Тепловые карты для значений левой и правой границы для признака y

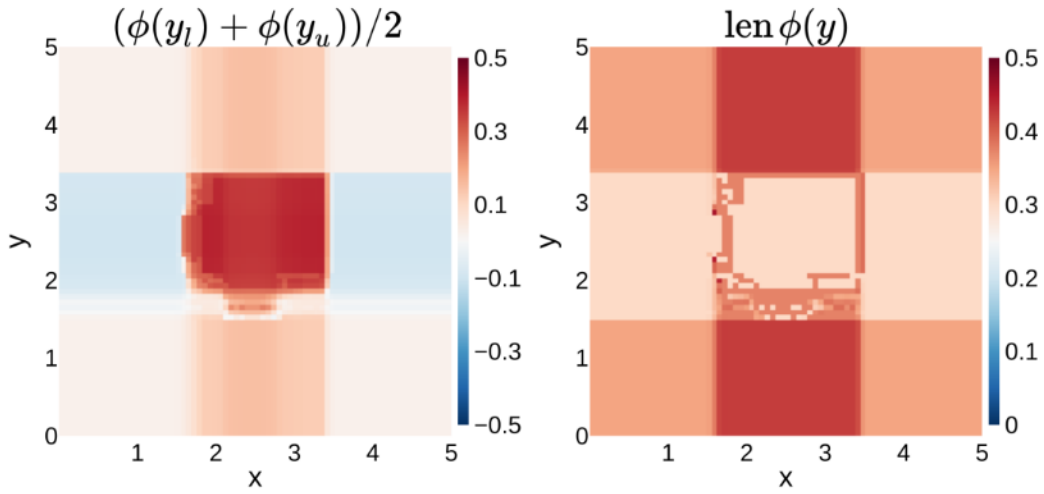

Рис. 4.7. Тепловые карты для значения середины и длины отрезка для признака y

Рассмотрим рисунки для значений середин отрезков чисел Шепли для признаков x и y. Во-первых, полученные рисунки действительно напоминают окружность. Конечно, на графике имеются некоторые прямоугольные области, но это связано с спецификой работы алгоритма Random Forest, в узлах которого находятся условия вида: $x > c$ или $y > c$.

Если рассмотреть рисунок середины значения отрезка ϕ для признака x, то для точек, у которых $y \in [1.5, 3.5]$ значения $\phi(x)$ больше, чем у точек, у которых $y \notin [1.5, 3.5]$. Действительно, ведь у точек с $y \in [1.5, 3.5]$, при изменении значения x может произойти изменение класса, поэтому признак x для них важен. В то же время, если у точек с $y \notin [1.5,3.5]$ изменять значение признака x, то смена класса не произойдет, и поэтому для таких точек $\phi(x)$ имеет значение по модулю близкое к нулю.

Точно такое же рассуждение можно провести и для $\phi(y)$. Точки с $x \in [1.5, 3.5]$ имеют более высокое значение $\phi(y)$ по модулю, так как для них при изменении y может произойти изменение класса. Точки с $x \notin [1.5,3.5]$ имеют меньшее по модулю значение $\phi(y)$ так, как изменение класса при изменении значения y для них произойти не может.

Также, видно, что внутри окружности важны оба признака. Это можно объяснить тем, что для того, чтобы проверить лежит ли точка внутри окружности

необходимо знать значения обеих координат. При этом, если одна из координат очень далеко находится от центра, то достаточно знать только ее значение, чтобы понять, что точка не лежит внутри окружности. Этим можно объяснить одинаковое значение $\phi(x)$ и $\phi(y)$ в углах квадрата.

Теперь рассмотрим тепловые карты для признаков x и y после сужения.

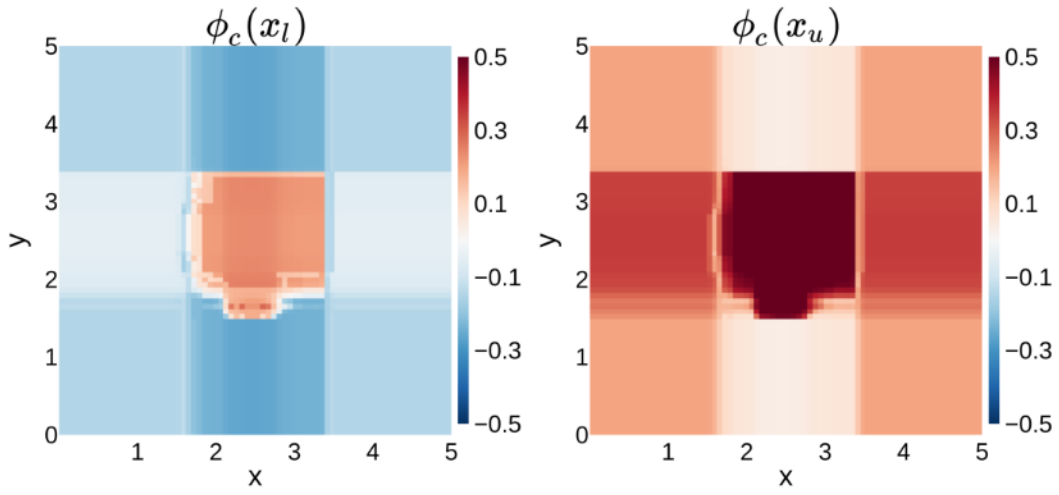

Рис. 4.8. Тепловые карты для значений левой и правой границы для признака x после сужения.

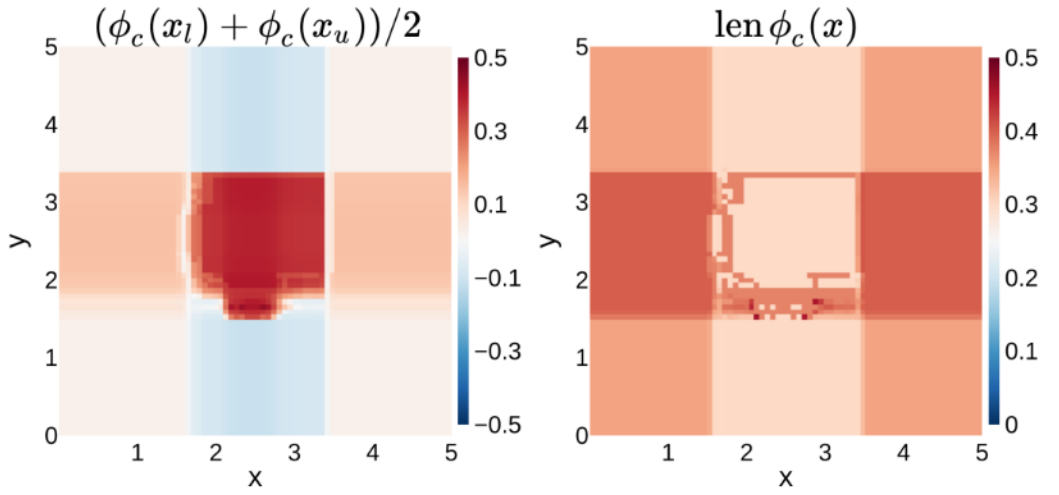

Рис. 4.9. Тепловые карты для значений середины и длины отрезка для признака x после сужения

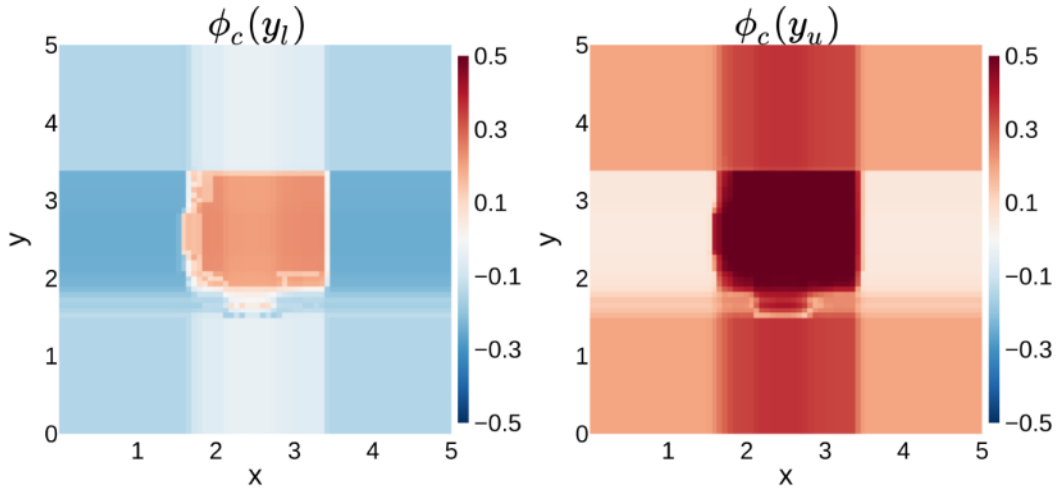

Рис. 4.10. Тепловые карты для значений левой и правой границы для признака y после сужения

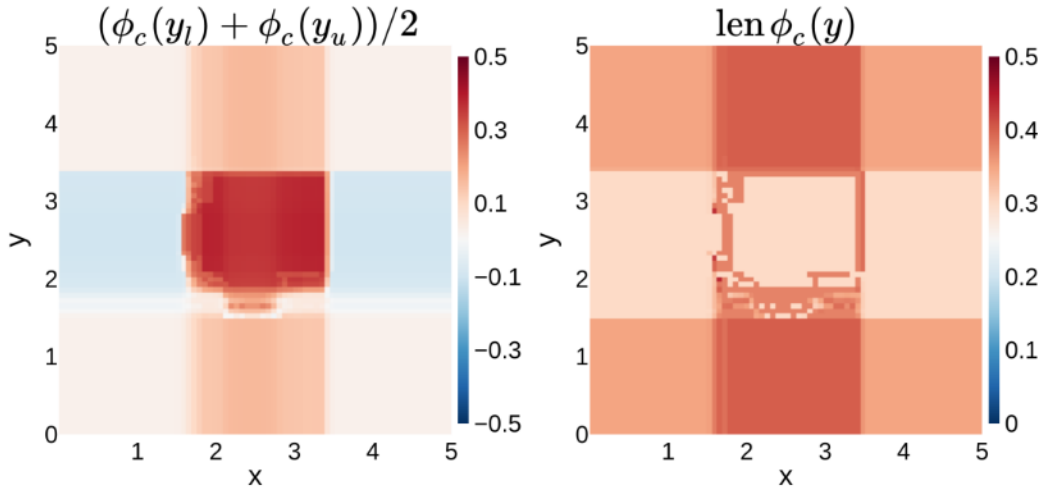

Рис. 4.11. Тепловые карты для значения середины и длины отрезка для признака y после сужения

Из рисунка для признака x видно, что были сужены значения границ для точек, у которых, $x \notin [1.5,3.5]$, а $y \in [1.5, 3.5]$. Длина отрезка для этих точек уменьшилась с 0.42564 до 0.401584. Для признака y произошло сужение для точек, у которых $y \notin [1.5,3.5]$, а $x \in [1.5, 3.5]$. Для них длина отрезка уменьшилась с 0.424544 до 0.401587. Таким образом, для некоторого набора точек получилось немного сузить полученные отрезки. Далее, рассмотрим зависимость изменения длин отрезков при изменении параметра ϵ. Ниже

представлены графики, иллюстрирующие такую зависимость для нескольких точек на плоскости.

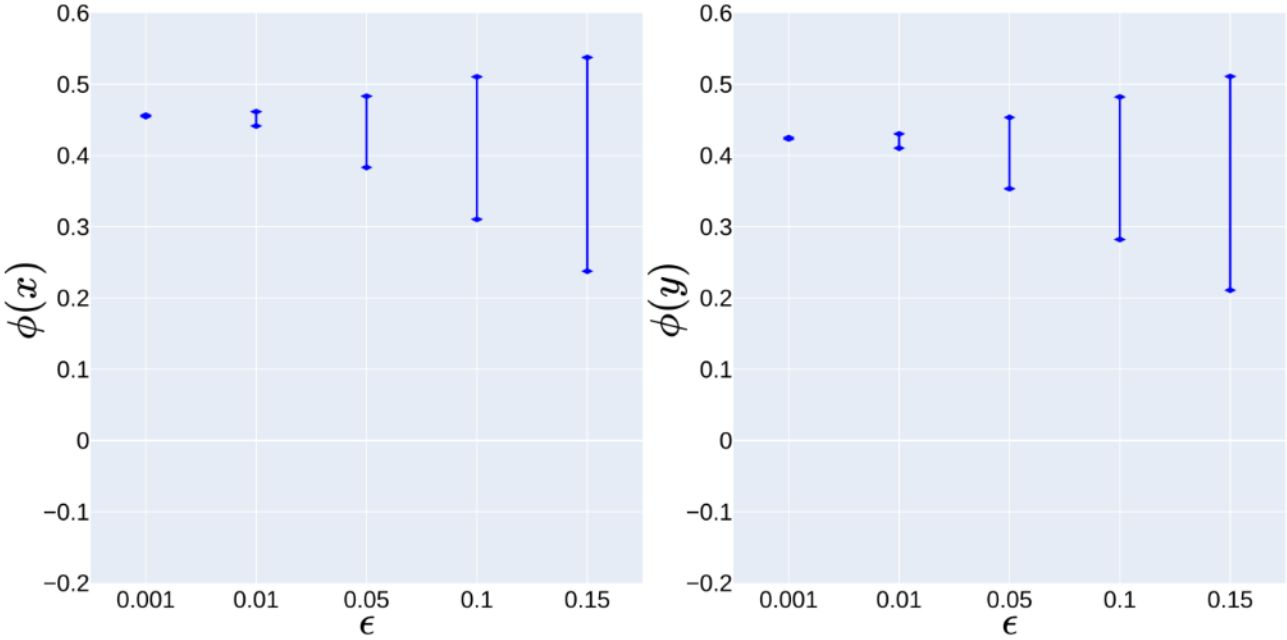

Рис. 4.12. Изменение границ отрезка в зависимости от параметра ϵ для признаков x и y для точки (2.5, 2.5)

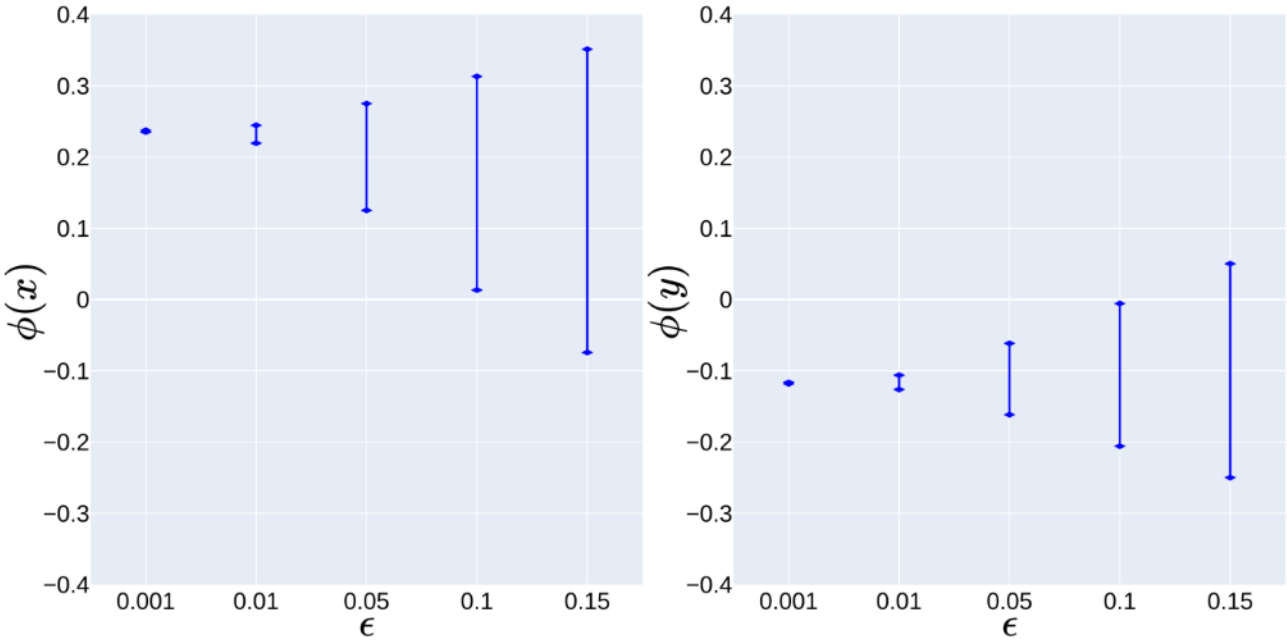

Рис. 4.13. Изменение границ отрезка в зависимости от параметра ϵ для признаков x и y для точки (3.6, 2.5)

На графиках видно, как уменьшается длина отрезка при уменьшении значения параметра ϵ. Также видно, что отрезки не симметричны относительно центра. Для точки (2.5, 2.5) отрезки получились почти идентичными. Это можно объяснить тем, что для точек в центре окружности важны значения обоих признаков. Для точки (3.6, 2.5) получилось, что признак x является более важным, чем признак y. Действительно, при изменении значения признака x, происходит смена класс, а при изменении значения y этого не происходит.

4.3.2. Датасет Окружность с тремя классами

Сначала рассмотри полученные тепловые карты для этого набора данных.

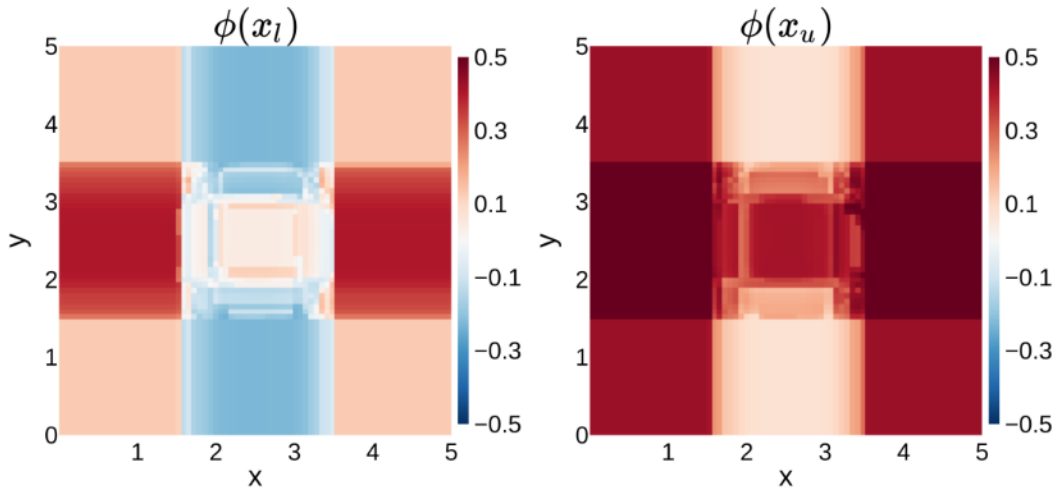

Рис. 4.14. Тепловые карты для значений левой и правой границы для признака x

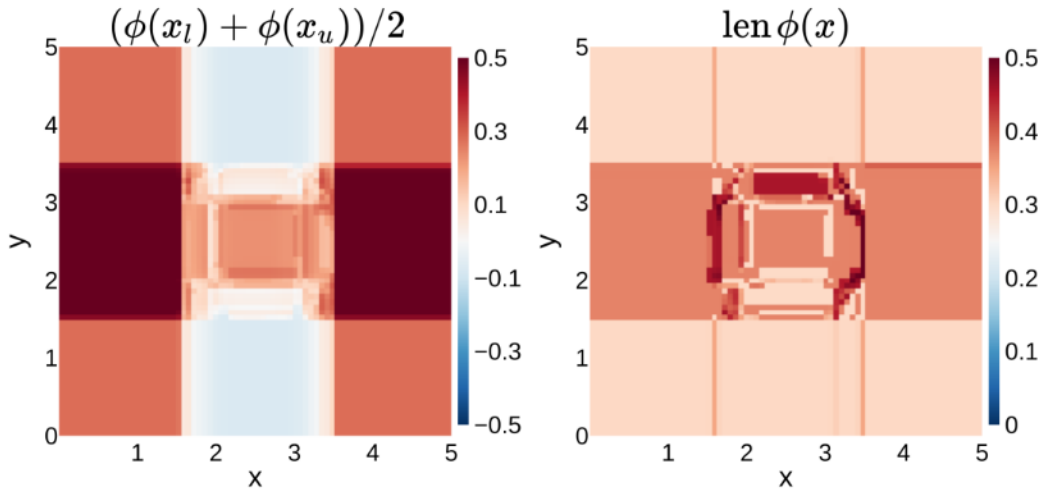

Рис. 4.15. Тепловые карты для значения середины и длины отрезка для признака *x*

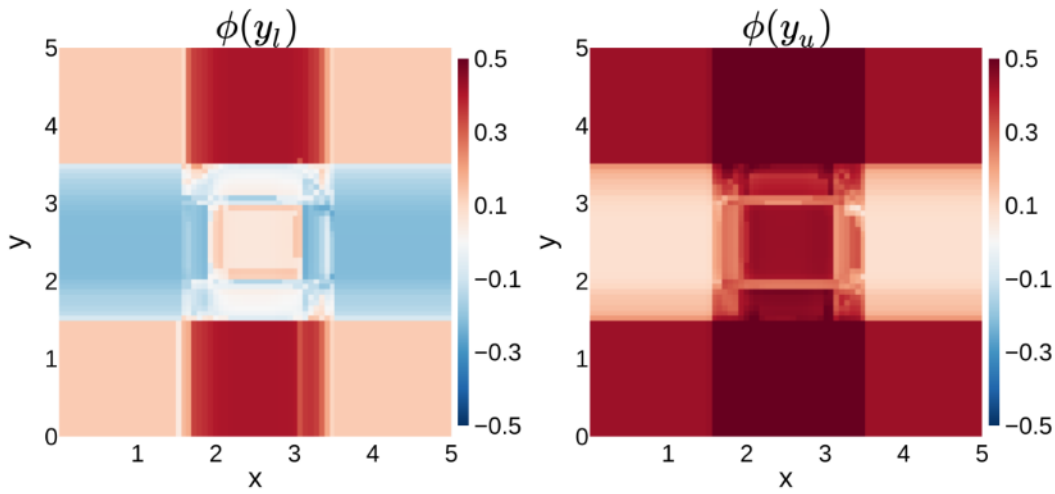

Рис. 4.16. Тепловые карты для значений левой и правой границы для признака *y*

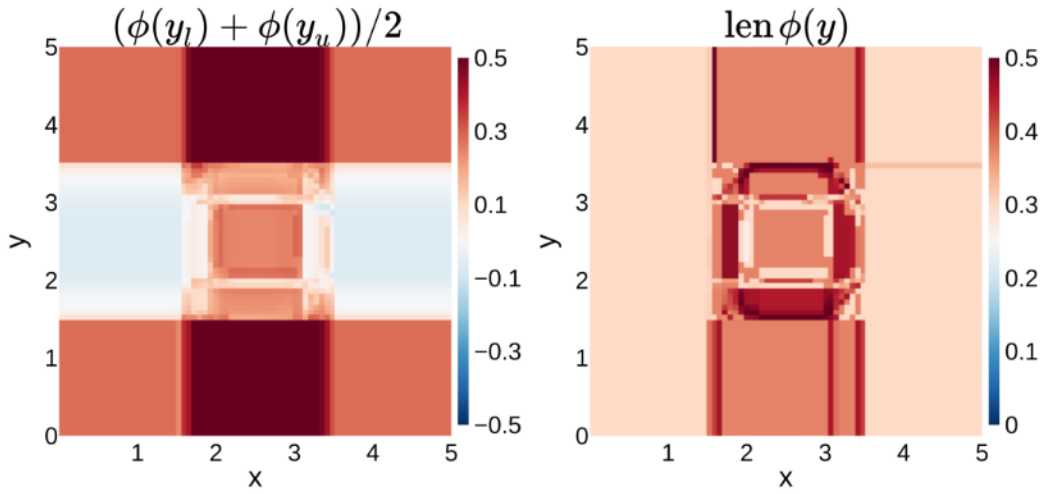

Рис. 4.17. Тепловые карты для значения середины и длины отрезка для признака y

Полученные рисунки напоминают рисунки для окружности с двумя классами. Стоит отметить, что отчетливо видны очертания дуги второй окружности. Видно, как меняются значения ϕ при переходе через границы маленькой и большой окружности. В углах значения $\phi(x)$ и $\phi(y)$ по-прежнему имеют одинаковое значение. Можно провести аналогичные рассуждения про изменение значений $\phi(x), \phi(y)$ при изменении значений одного из признаков, что и для случая с одной окружностью. Ниже, представлены графики зависимостей изменения границ отрезка при изменении параметра ϵ для точек (2.5, 2.5) и (4.5, 4.5).

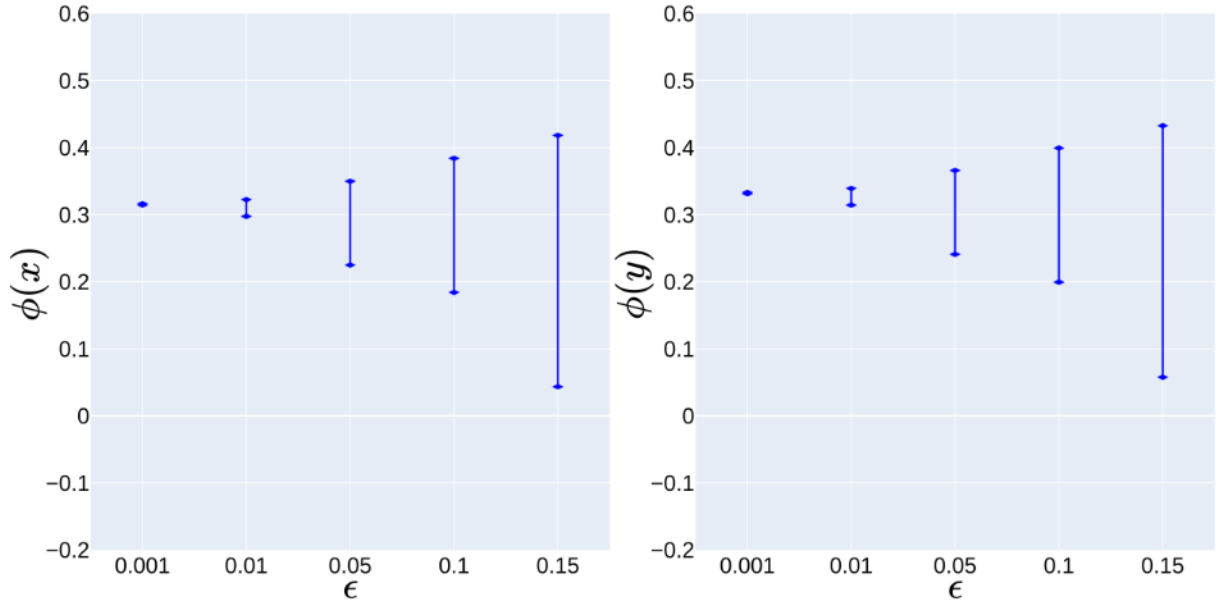

Рис 4.18. Изменение границ отрезка в зависимости от параметра ϵ для признаков x и y для точки (2.5, 2.5)

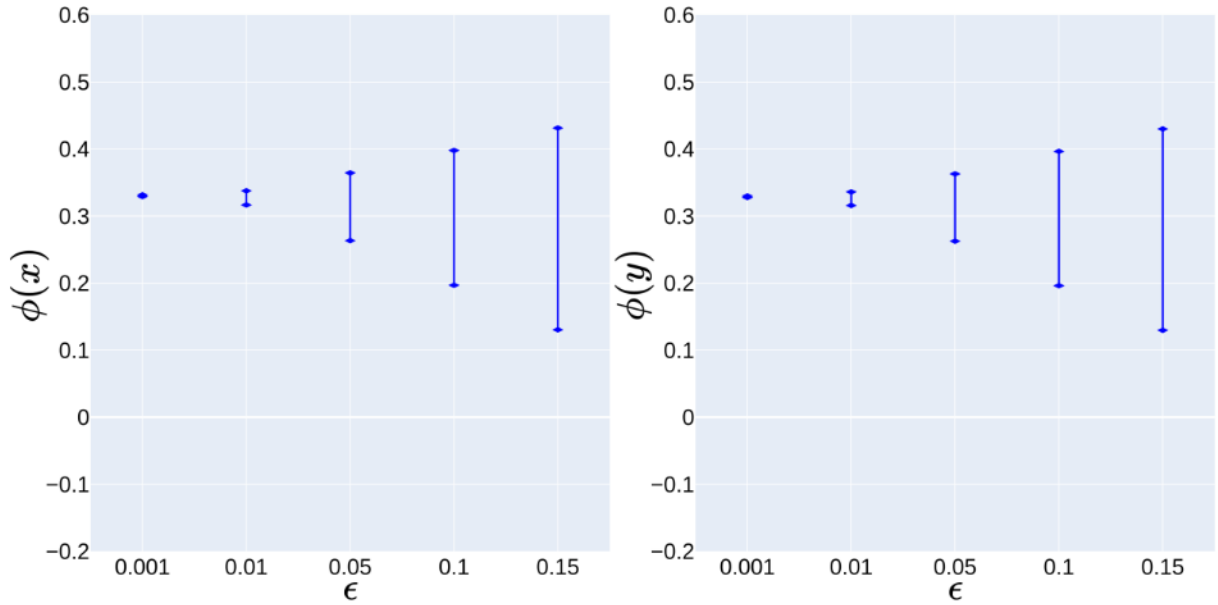

Рис 4.19. Изменение границ отрезка в зависимости от параметра ϵ для признаков x и y для точки (4.5, 4.5)

Значимость признаков совпала для обеих точек, однако для точки (4.5, 4.5) отрезки получились немного более узкими, чем для точки (2.5, 2.5), и это можно объяснить тем, для этих точек различались распределения вероятностей, которые предсказала модель. Для точки (4.5, 4.5) модель была более уверенна в

своем предсказании, и как следствие со 100% вероятностью выбрала один из классов. Для точки (2.5, 2.5) модель была менее уверенна в своем предсказании, и выбрала максимальный класс с вероятностью 97%.

4.3.3. Датасет Четыре кластера данных

Рассмотрим построенные тепловые карты для этого набора данных.

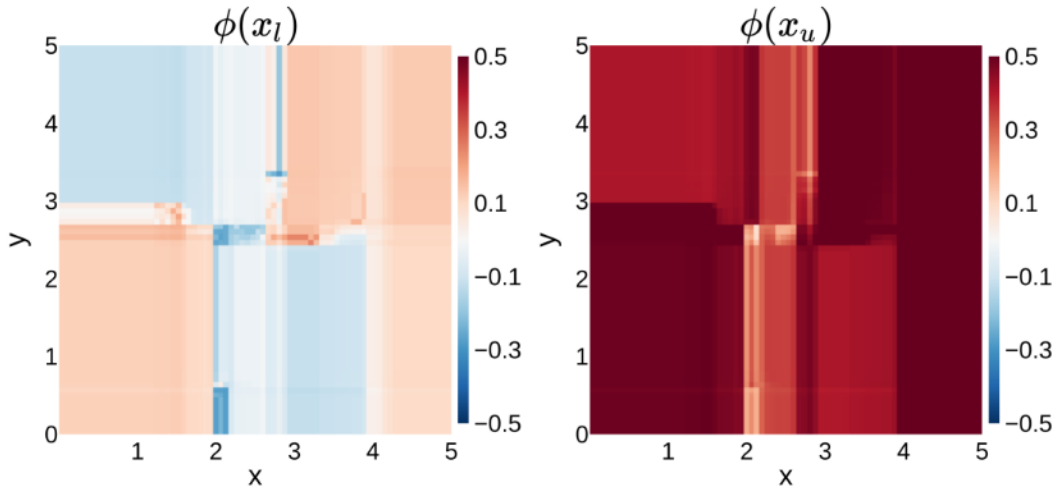

Рис 4.20. Тепловые карты для значений левой и правой границы числа Шепли для признака x

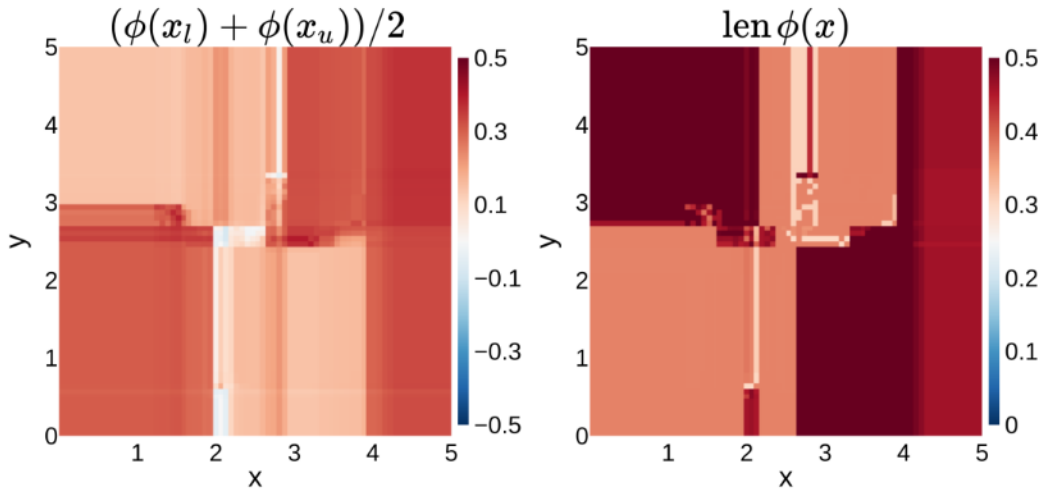

Рис 4.21. Тепловые карты для значения середины отрезка чисел Шепли и длины этого отрезка для признака *x*

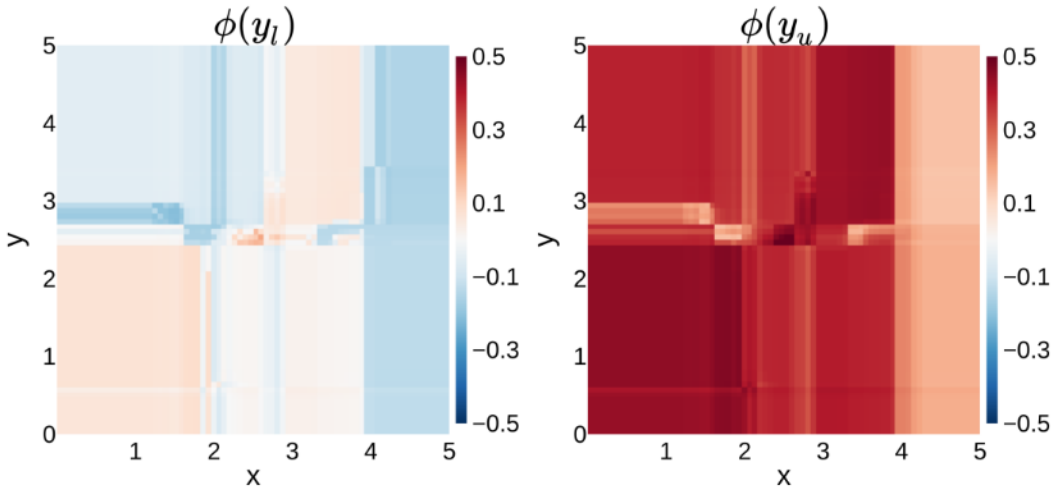

Рис. 4.22. Тепловые карты для значений левой и правой границы числа Шепли для признака y

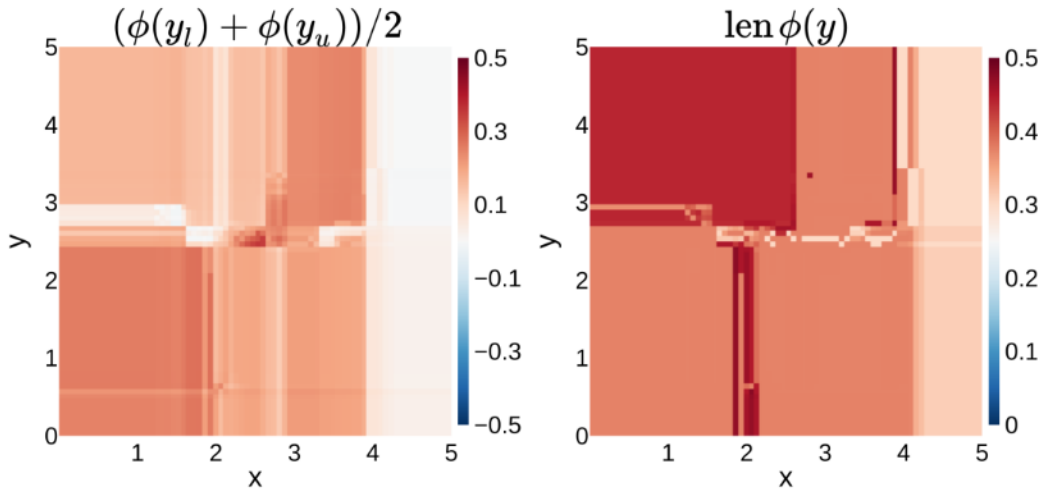

Рис. 4.23. Тепловые карты для значения середины отрезка чисел Шепли и длины этого отрезка для признака *y*

На полученных тепловых картах отлично видно разделение плоскости на четыре части, каждая из которых соответствует одному из четырех кластеров данных. Исследуем связь между значениями чисел Шепли и уверенностью предсказаний модели. Если рассмотреть тепловые карты, где цветом отображается предсказанный класс (рис. 5.32. (а)), и максимальная вероятность из предсказанного распределения (рис. 5.32. (б)), то будет заметна схожесть с представленными выше тепловыми картами для чисел Шепли.

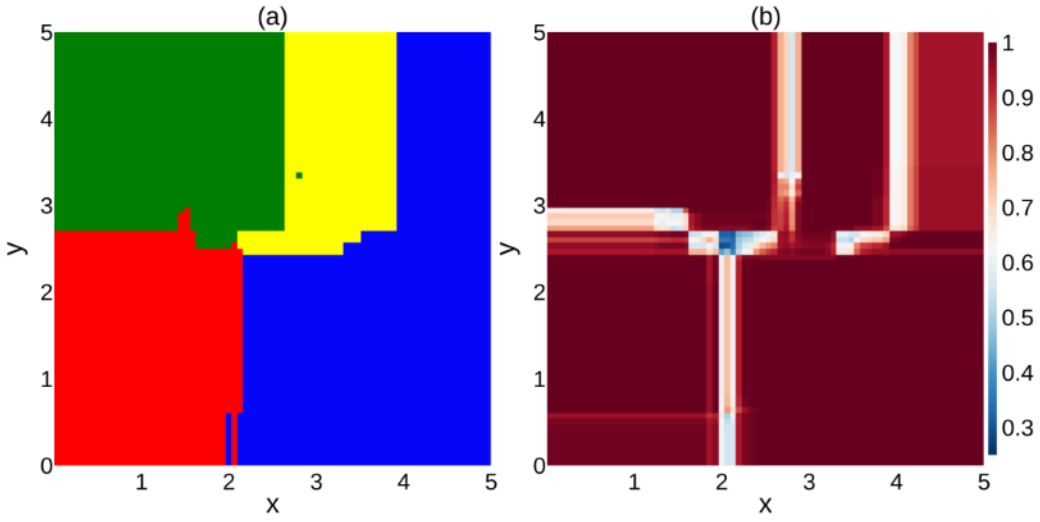

Рис. 4.24. Предсказания модели. Слева (a) представлен предсказанный класс: красный – класс 0, зеленый – класс 1, синий –класс 2, желтый – класс 3. Справа (b) представлена максимальная вероятность из распределения.

Из этих графиков, видно, что значения чисел Шепли по модулю близкие к нулю получились на границах разных классов, а значение большие по модулю получились в местах, где по близости находятся точки, принадлежащие только одному из классов.

4.3.4. Datacem Seeds

Этот набор данных является реальным и взят из UCI Machine Learning Repository. В нем имеется 7 признаков, чьи значения являются действительными числами. Целевая переменная может принимать три значения, таким образом разработанная модель, чьи предсказания требуется объяснить решает задачу классификации на три класса. Для построения графиков были взяты точки со значениями, которые указаны в таблице 5.1.

Таблица 5.1

Значения признаков

	Признаки						
Номер точки	area	per	comp	len_ker	width	asym	len_gr
1	18.14	16.12	0.8772	6.059	3.563	3.619	6.011
2	16.68	14.69	0.8703	6.357	3.465	2.526	5.603

На графиках синим цветом изображены полученные отрезки при $\epsilon = 0.15$. Также красным кругом отмечено значение числа Шепли для заданного признака при $\epsilon = 0$.

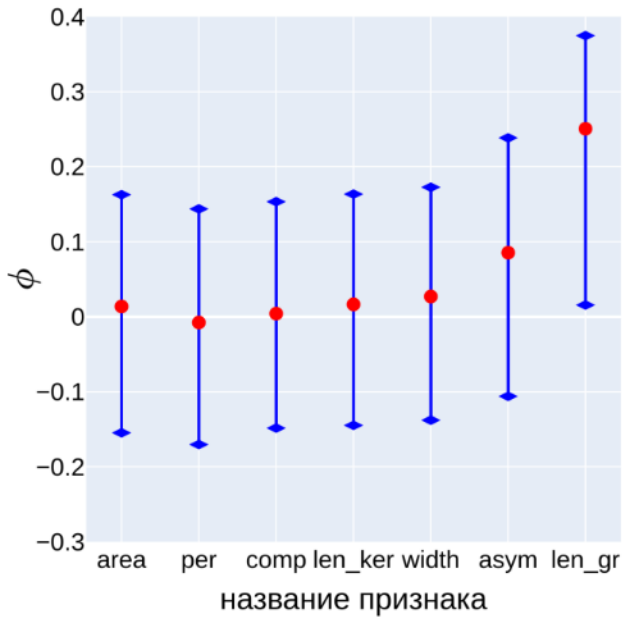

Рис. 4.25. Полученные отрезки для точки №1

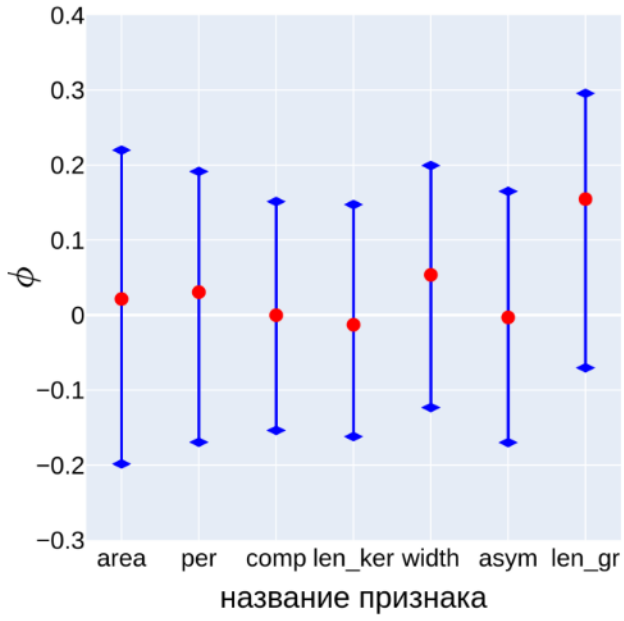

Рис. 4.26. Полученные отрезки для точки №2

Видно, что на обоих графиках есть признаки, у которых центр отрезка расходится со значением числа Шепли, полученным при $\epsilon = 0$. Более того, видно, что величина этого расхождения коррелирует со значением числа Шепли: чем больше значение числа Шепли по модулю, тем больше расхождение между серединой отрезка и значением для $\epsilon = 0$. Если смотреть на значения, полученные при $\epsilon = 0$, то в обоих примерах можно сделать однозначные выводы о том, какие признаки являются важными, а какие нет. Однако, если рассматривать отрезки, так как на обоих примерах отрезки для каждого из признаков имеют большую область пересечения, может получиться так, что истинные значения чисел Шепли будут совершенно другими, по сравнению со случаем при $\epsilon = 0$.

В то же время, обычный SHAP для каждого примера выдает таблицу размерности $n \times m$, где n – количество признаков, а m – количество классов, где каждая ячейка (i, j) хранит значение посчитанного числа Шепли для вероятности, полученной для класса под номером j, и признака под номером i. Ниже приведена такая таблица, посчитанная для точки №1.

Таблица 5.2

Числа Шепли, полученные с помощью обычного SHAP

	Класс 1	Класс 2	Класс 3
area	-0.0195	0.1079	-0.0884
per	-0.0358	0.1090	-0.0732
comp	0.0051	0.0010	-0.0061
len_ker	-0.0803	0.1281	-0.0478
width	0.0113	0.0399	-0.0512
asym	-0.0235	0.0179	0.0056
len_gr	-0.2045	0.2540	-0.0495

По сравнению с графиками, представленными выше эту таблицу очень тяжело интерпретировать из-за большого количества значений в ней. Таким образом, для задачи классификации результаты, предоставленные

модификацией SHAP, описанной в данной работе, являются более интерпретируемыми, чем результаты обычного SHAP.

4.3.5. Датасет Ecoli

Еще один реальный датасет, взятый из UCI Machine Learning Repository, который содержит 7 признаков, чьи значения являются действительными числами. Целевая переменная может принимать восемь различных значений, таким образом объясняемая модель решает задачу классификации на восемь классов. Построим графики, отображающие полученные отрезки при $\epsilon = 0.15$ и значение, полученное при $\epsilon = 0$. Для построения графиков были взяты точки со следующими значениями признаков:

Таблица 5.3

Значения признаков

	Признаки						
Номер точки	mcg	gvh	lip	chg	aac	alm1	alm2
1	0.799	0.826	0.650	0.595	0.169	0.452	0.851
2	0.517	0.790	0.829	0.596	0.834	0.578	0.323

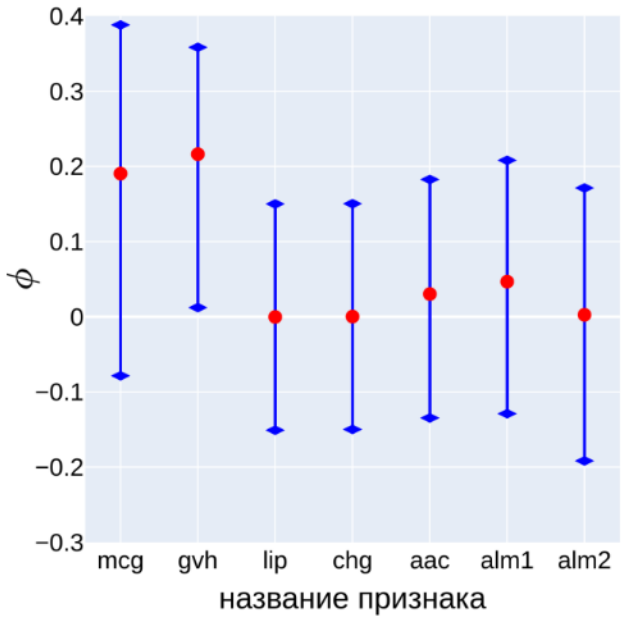

Рис. 4.27. Полученные отрезки для точки №1

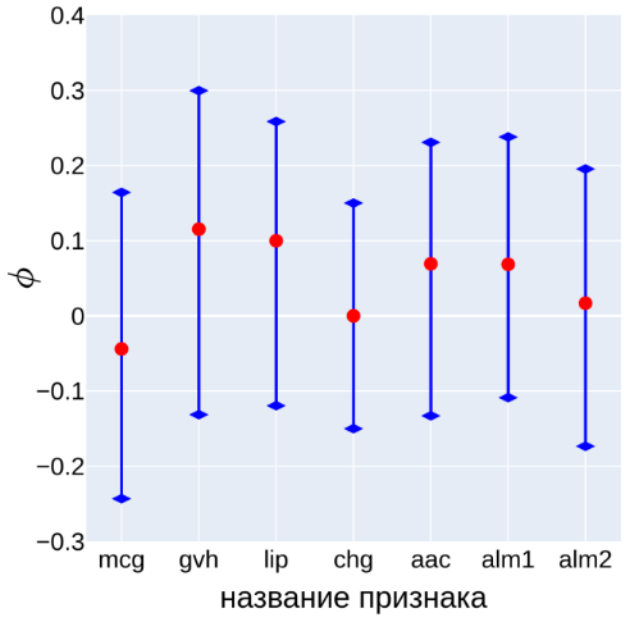

Рис. 4.28. Полученные отрезки для точки №2

Полученные рисунки иллюстрируют ценность полученных отрезков. Для точки №1, если смотреть на значения, полученные для признаков “mcg” и “gvh” при $\epsilon = 0$, то получается, что признак “gvh” является более важным, чем признак “mcg”. В то же время, верхняя граница для признака mcg имеет большее значение, чем у признака “gvh”, тем самым это говорит о том, что признак mcg может быть более важным, чем признак “gvh”. Этого нельзя было бы понять, если просто смотреть на точки, полученные при $\epsilon = 0$. Если рассмотреть значения, полученные для этих же двух признаков при $\epsilon = 0$ для точки №2, то получиться, что признак “gvh” является наиболее важным, а признак “mcg” наименее важным. Однако, если рассмотреть отрезки, построенные для этих признаков, то они будут пересекаться приблизительно на две трети своих длин, таким образом может оказаться что признак “mcg” будет даже более важным, чем признак “gvh”.

4.3.6. Датасет Glass Identification

Этот набор данных также был взят из UCI Machine Learning Repository. Он содержит десять различных признаков, каждый пример принадлежит к одному из восьми классов. Ниже представлены графики для точек из таблицы 5.4.

Таблица 5.4

Значения признаков

	Признаки								
Номер точки	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
1	1.536	16.324	2.146	0.560	72.485	3.602	13.142	0.068	0.005
2	1.515	13.369	0.632	1.627	74.874	4.543	6.624	0.067	0.025

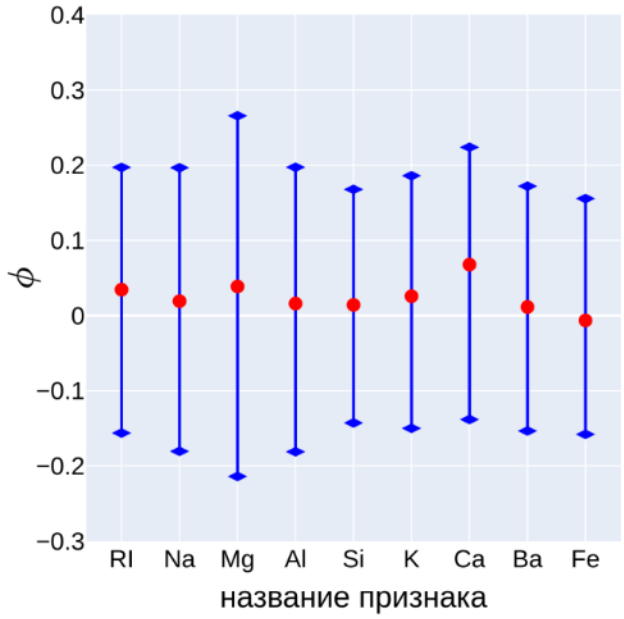

Рис. 4.29. Полученные отрезки для точки №1

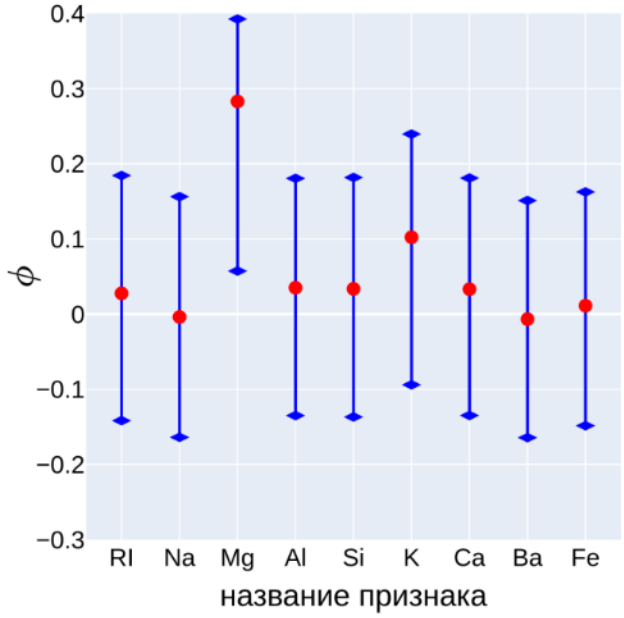

Рис. 4.30. Полученные отрезки для точки №2

На графике для первой точки видна ситуация, когда если смотреть на значения при $\epsilon = 0$, то выходит, что самым важным является признак “Ca”. Однако, если рассматривать отрезки значений чисел Шепли, то выходит, что признак “Mg” может быть более важен. Для второй точки, на графике видно, что признаки “Mg” и “K” являются самыми важными, но если рассматривать границы отрезков, то возможна ситуация, когда эти признаки являются наименее значимыми.

ЗАКЛЮЧЕНИЕ

В результате выполнения этой работы была достигнута поставленная цель, а именно была разработана и реализована версия алгоритма SHAP для случая многоклассовой классификации с использованием модели ϵ-засорения. Также были выполнены следующие задачи:

- исследованы методы интерпретируемого машинного обучения;
- сгенерированы и подготовлены наборы данных, необходимых для проведения экспериментов;
- проведено тестирование разработанного алгоритма, а также эксперименты на синтетических и реальных данных;
- проведен анализ результатов экспериментов.

Так как метод SHAP является универсальным методом интерпретируемого машинного обучения и способен работать с моделями, которые используют разные типы данных, то его реализация для случая многоклассовой классификации, является очень важным дополнением к уже существующим методам. Еще одной очень важной частью этой работы является то, что разработанная реализация позволяет учитывать возможную неточность предсказания моделей. Это достигается за счет того, что при вычислении значений чисел Шепли, вместо единственного распределения вероятностей предсказания модели, рассматривается множество распределений вероятностей, построенное в соответствии с моделью ϵ-засорения. Таким образом, вместо единственного значения числа Шепли, получается отрезок значений, имеющий нижнюю и верхнюю границу. Для задач многоклассовой классификации, отрезки, которые получаются на выходе легче интерпретировать, чем большую таблицу, которую выдает стандартная версия SHAP.

В работе были представлены примеры, которые показывают полезность рассмотрения отрезков значений вместо одиночных значений чисел Шепли. Одиночные значения, позволяют сделать однозначные выводы о важности

признаков, в то время как отрезки значений хоть и коррелируют с одиночными значениями, но также показывают, что истинные значения чисел Шепли могут быть значительно отличными от одиночных.

В качестве дальнейшего направления исследований можно выделить два момента: ускорение работы алгоритма и уменьшение неопределенности, возникающей при вычислении чисел Шепли. Ускорение алгоритма является очень важной задачей, так как это позволит применять его для объяснения моделей, которые обучены на датасетах, которые содержат большое число признаков. Как раз для таких датасетов, проблема возникающей неопределенность при вычислении чисел Шепли является наиболее актуальной, так как при увеличении числа признаков, экспоненциально увеличивается число слагаемых в сумме, а как следствие и количество вычислений $f(S)$ на неполном наборе признаков. Так как значения опускаемых признаков заменяются на константные значения, которые зачастую плохо аппроксимируют истинные значения, то это ведет к тому, что полученные значения чисел Шепли лежат далеко от настоящих значений, что, в свою очередь, ведет к неверным выводам при интерпретации предсказаний.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

DeepLift – Deep Learning Important FeaTures

Grad-CAM – Gradient-weighted Class Activation Mapping

LIME – Local Interpretable Model-Agnostic Explanations

SHAP – SHapley Additive exPlanations

SVM – Support Vector Machine

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ahmad M. A., Teredesai A., Eckert C. Interpretable machine learning in healthcare Institute of Electrical and Electronics Engineers Inc., 2018.C. 447.

2. Brigo D. [и др.]. Interpretability in deep learning for finance: a case study for the Heston model // SSRN Electronic Journal. 2021.

3. Bussmann N. [и др.]. Explainable Machine Learning in Credit Risk Management // Computational Economics. 2021. № 1 (57). С. 203–216.

4. Chen J., Li S. E., Tomizuka M. Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning // IEEE Transactions on Intelligent Transportation Systems. 2021.

5. Fisher A., Rudin C., Dominici F. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously // Journal of Machine Learning Research. 2018. (20).

6. Holzinger A. Explainable AI and Multi-Modal Causability in Medicine // i-com. 2021. № 3 (19). С. 171–179.

7. Kim J., Canny J. Interpretable Learning for Self-Driving Cars by Visualizing Causal Attention // Proceedings of the IEEE International Conference on Computer Vision. 2017. (2017-October). С. 2961–2969.

8. Lundberg S., Lee S.-I. A Unified Approach to Interpreting Model Predictions // Advances in Neural Information Processing Systems. 2017. (2017-December). С. 4766–4775.

9. Miller T. Explanation in artificial intelligence: Insights from the social sciences // Artificial Intelligence. 2019. Т. 267. С. 1–38.

10. Moeyersoms J. [и др.]. Explaining Classification Models Built on High-Dimensional Sparse Data 2016.

11. Ribeiro M. T., Singh S., Guestrin C. "Why should I trust you?" Explaining the predictions of any classifier Association for Computing Machinery, 2016.C. 1135–1144.

12. Selvaraju R. R. [и др.]. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization // International Journal of Computer Vision. 2016. № 2 (128). С. 336–359.

13. Shapley L. S. 17. A Value for n-Person Games Princeton University Press, 2016.С. 307–318.

14. Shrikumar A., Greenside P., Kundaje A. Learning Important Features Through Propagating Activation Differences // 34th International Conference on Machine Learning, ICML 2017. 2017. (7). С. 4844–4866.

15. Subramanian A. [и др.]. SPINE: SParse Interpretable Neural Embeddings // 32nd AAAI Conference on Artificial Intelligence, AAAI 2018. 2017. С. 4921–4928.

16. Vellido A. The importance of interpretability and visualization in machine learning for applications in medicine and health care // Neural Computing and Applications. 2020. № 24 (32). С. 18069–18083.