Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *Робототехника и комплексная автоматизация*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

НА ТЕМУ

«Интерпретация аппроксимирующих моделей машинного обучения»

Студент РК6-82Б — (Группа) — (подпись, дата) — А.В. Конов (инициалы и фамилия)

Руководитель ВКР — (подпись, дата) — А.П. Карпенко (инициалы и фамилия)

Нормоконтролер — (подпись, дата) — С.В. Грошев (инициалы и фамилия)

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой *РК-6*
(индекс)

_____ *А.П. Карпенко*
(инициалы и фамилия)

«___» _____ *2023* г.

З А Д А Н И Е
на выполнение выпускной квалификационной работы

Студент группы РК6-82Б

Конов Антон Викторович
(фамилия, имя, отчество)

Тема выпускной квалификационной работы

Интерпретация аппроксимирующих моделей машинного обучения

Источник тематики (кафедра, предприятие, НИР): кафедра

Тема квалификационной работы утверждена распоряжением по факультету РК № _____ от «___» _____ *2023* г.

Часть 1. *Введение в предметную область*

В рамках введения в предметную область должны быть изучены методы интерпретации машинного обучения. Должна быть обоснована актуальность исследований.

Часть 2. *Математическая постановка задачи*

Должна быть изучена методика интерпретации моделей машинного обучения на основе решающих деревьев.

Часть 3. *Проведение вычислительных экспериментов*

Вычислительные эксперименты по интерпретации аппроксимирующих моделей машинного обучения должны быть проведены с использованием разработанного программного обеспечения.

Оформление выпускной квалификационной работы:

Расчетно-пояснительная записка на 63 листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

21 рисунок, 0 таблиц, 28 источников, 16 графических листов
Слайд 1. Введение
Слайд 2. Цели и задачи
Слайд3. Модели машинного обучения

Слайд 4. Методы интерпретации моделей машинного обучения
Слайд 5. Метод LIME
Слайд 6. Метод PDP
Слайд 7. Метод SHAP
Слайд 8. Блок-схема разработанного класса Inter
Слайд 9. Вычислительные эксперименты
Слайд 10. Эксперимент 1
Слайд 11. Эксперимент 1
Слайд 12. Эксперимент 1
Слайд 13. Эксперимент 2
Слайд 14. Эксперимент 2
Слайд 15. Эксперимент 2
Слайд 16. Заключение

Дата выдачи задания «13» февраля 2023 г.

Студент А.В. Конов

(Подпись, дата) (И.О.Фамилия)

Руководитель выпускной квалификационной работы А.П. Карпенко

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **РК**
КАФЕДРА **РК-6**
ГРУППА **РК6-82Б**

УТВЕРЖДАЮ
Заведующий кафедрой *РК-6*
(индекс)

А.П. Карпенко
(инициалы и фамилия)

«____» _____ *2023* г.

КАЛЕНДАРНЫЙ ПЛАН
выполнения выпускной квалификационной работы

студента: *Конов Антон Викторович*
(Фамилия, имя, отчество)

Тема выпускной квалификационной работы: Интерпретация аппроксимирующих моделей машинного обучения

№ п/п	Наименование этапов выпускной квалификационной работы	Сроки выполнения		Отметка о выполнении	
		план	факт	Должность	ФИО, подпись
1.	Задание на выполнение работы. Формулирование проблемы, цели и задач работы	*13.02.2023 Планируемая дата*		Руководитель ВКР	*А.П. Карпенко*
2.	1 часть: *введение в предметную область*	*18.02.2023 Планируемая дата*		Руководитель ВКР	*А.П. Карпенко*
3.	Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач	*28.02.2023 Планируемая дата*		Заведующий кафедрой	*А.П. Карпенко*
4.	2 часть: *математическая постановка задачи*	*31.03.2023 Планируемая дата*		Руководитель ВКР	*А.П. Карпенко*
5.	3 часть: *проведение вычислительных экспериментов*	*30.04.2023 Планируемая дата*		Руководитель ВКР	*А.П. Карпенко*
6.	1-я редакция работы	*31.05.2023 Планируемая дата*		Руководитель ВКР	*А.П. Карпенко*
7.	Подготовка доклада и презентации	*07.06.2023 Планируемая дата*			
8.	Заключение руководителя	*08.06.2023 Планируемая дата*		Руководитель ВКР	*А.П. Карпенко*
9.	Допуск работы к защите на ГЭК (нормоконтроль)	*09.06.2023 Планируемая дата*		Нормоконтролер	*С.В. Грошев*
10	Внешняя рецензия	*07.06.2023 Планируемая дата*			
11	Защита работы на ГЭК	*19.06.2023 Планируемая дата*			

Студент _____ *А.В. Конов*
(подпись, дата)

Руководитель работы _____ *А.П. Карпенко*
(подпись, дата)

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

НАПРАВЛЕНИЕ
НА ЗАЩИТУ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Председателю
Государственной Экзаменационной Комиссии № _____

факультета «Робототехника и комплексная автоматизация» МГТУ им. Н.Э. Баумана

Направляется студент *Конов Антон Викторович* группы *РК6-82Б*

на защиту выпускной квалификационной работы *Интерпретация аппроксимирующих моделей машинного обучения*
Декан факультета _____ «____» _____2023 г.

Справка об успеваемости

Студент *Конов Антон Викторович* за время пребывания в МГТУ имени Н.Э. Баумана

с *2019* г. по *2023* г. полностью выполнил учебный план со следующими оценками: отлично – ____, хорошо – ____, удовлетворительно – ____ .

Инспектор деканата _____

Отзыв руководителя выпускной квалификационной работы

Студент Фамилия И.О. в процессе выполнения ВКР проявил себя как … Результаты, полученные в процессе реализации задания, позволили сделать вывод о … целесообразности/нецелесообразности выбранных путей решения поставленной задачи, … невозможности применения … Автором были получены результаты, обладающие научной новизной… Работа выполнена автором самостоятельно, в полном объёме, в полном соответствии с заданием и календарным планом.

Выполненная работа соответствует заявленной теме, а также требованиям, предъявляемым к выпускным квалификационным работам, и заслуживает оценки «хорошо», а ее автор – присуждения степени магистр техники и технологий по направлению 09.03.01 – «Информатика и вычислительная техника».

Руководитель ВКР _____ А.П. Карпенко «____» _____ 2023 г.
(подпись) (ФИО) (дата)

Студент _____ А.В. Конов «____» _____ 2023 г.
(подпись) (ФИО) (дата)

РЕФЕРАТ

Работа посвящена разработке программного обеспечения, автоматизирующего процесс интерпретации аппроксимирующих моделей машинного обучения. Интерпретация моделей машинного обучения - это процесс анализа и объяснения принятия решений моделью на основе входных данных и внутренней структуры модели, с целью получения понимания ее работы и принципов прогнозирования.

В рамках работы был разработан класс, автоматизирующий процесс интерпретации аппроксимирующих моделей машинного обучения. В главе 1 произведен обзор моделей машинного обучения, обзор методов интерпретации и обзор программного обеспечения (ПО) методов интерпретации. В главе 2 рассмотрена методика интерпретации моделей машинного обучения на основе решающих деревьев. В главе 3 описана программная реализация разработанного программного обеспечения, проведены вычислительные эксперименты и их анализ.

Расчётно-пояснительная записка содержит 63 страницы, 21 рисунок, 0 таблиц, 28 источников.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современном мире машинное обучение играет ключевую роль во многих областях [1]. С использованием большого количества данных и мощных вычислительных ресурсов, модели машинного обучения могут давать высокоточные прогнозы. Однако, часто возникает необходимость не только в предсказании, но и в понимании процессов, лежащих в основе принятия решений модели. Именно в этой ситуации применяются методы интерпретации моделей машинного обучения [2].

Интерпретация методов машинного обучения - это процесс понимания того, как модель принимает решения и какие признаки оказывают наибольшее влияние на результат.

Для решения этой задачи было разработано множество методов интерпретации моделей машинного обучения. Они позволяют объяснить, какие признаки оказывают наибольшее влияние на результат, и, таким образом, улучшить понимание того, как модель принимает решения. Кроме того, интерпретация моделей позволяет выявлять необходимость изменения значений признаков для улучшения качества модели и предотвращения проблем с переобучением.

1. Введение в предметную область

1.1. Постановка задачи и основные понятия

Интерпретация моделей машинного обучения необходима для понимания того, как они принимают решения, и чтобы узнать, какие признаки вносят наибольший вклад в предсказание целевой переменной, что позволяет определить, какие факторы важны для конечного результата.

Основные термины, используемые в работе:

- признак;
- целевая переменная;
- входные данные;
- выходные данные;
- объект;
- набор данных.

Признак - это свойство или атрибут объекта, который используется для описания объекта в виде числовых или категориальных значений, которые затем используются для обучения модели.

Целевая переменная - это переменная, значение которой модель пытается предсказать на основе входных данных.

Входные данные - это набор признаков объекта, который используется для обучения модели и предсказания значения целевой переменной.

Выходные данные - это результат предсказания модели на основе входных данных, то есть значения целевой переменной, которые модель выдает в ответ на входные данные.

Объект - это элемент входных данных, который анализируется и на котором модель машинного обучения обучается.

Набор данных - собрание информации, организованной в структурированную форму, которая содержит различные типы данных и используется для обучения модели машинного обучения.

В рамках проводимой работы необходимо:

- изучить существующие модели машинного обучения, описать их преимущества и недостатки;
- изучить существующие подходы к интерпретации моделей машинного обучения, описать их преимущества и недостатки.

1.2. Обзор моделей машинного обучения

Модели машинного обучения— это класс алгоритмов, позволяющих компьютерным системам обучаться на основе данных, то есть строить предсказательные модели или классификаторы, опираясь на статистические свойства имеющихся входных данных [3]. В отличие от традиционных методов программирования, где задача решается напрямую, без учета данных, в машинном обучении процесс решения задачи основан на анализе данных, на основе которых строится математическая модель, позволяющая решать задачи классификации или регрессии на новых данных.

Задача классификации - задача машинного обучения, которая прогнозирует распределение элементов данных по классам, на основе его признаков. В этой задаче алгоритм обучается на основе размеченных входных данных, чтобы научиться отличать один класс от другого.

Задача регрессии - это задача машинного обучения, которая заключается в прогнозировании числового значения для целевой переменной на основе ее связи с другими признаками. В этой задаче алгоритм обучается на основе размеченных входных данных, чтобы научиться предсказывать целевую переменную на основе ее связи с другими признаками.

1.2.1. Классические модели машинного обучения

Классические модели машинного обучения - это модели обучения на основе статистических алгоритмов, которые используются для анализа и обработки данных [4].

Основные этапы построения классической модели машинного обучения:

1) загрузка данных и подготовка их для анализа;

2) разделение выборки на обучающую и тестовую;

3) обучение модели на обучающей выборке;

4) проверка качества работы модели на тестовой выборке.

Основные преимущества классических методов машинного обучения:

- модели просты в реализации и не требуют большого количества параметров;
- работают с категориальными и числовыми признаками;
- простота интерпретации моделей.

Основные недостатки классических методов машинного обучения:

- модели чувствительны к выбросам;
- модели не устойчивы к мультиколлинеарности;
- необходимость правильной подготовки данных.

Логистическая регрессия

Логистическая регрессия - это модель машинного обучения, которая используется для решения задач бинарной или многоклассовой классификации [5].

В логистической регрессии используется логистическая функция, которая преобразует значения линейной комбинации признаков в вероятность принадлежности к одному из классов. В результате, модель определяет вероятность принадлежности к одному из классов для каждого объекта, и выбирает тот класс, для которого вероятность наибольшая.

Преимущества модели логистической регрессии:

- показывает хорошие результаты на различных наборах данных, включая данные с большим числом признаков и небольшим количеством объектов;
- может быть адаптирована для решения многоклассовых задач классификации с использованием различных подходов.

Недостатком модели логистической регрессии является, то что логистическая регрессия не может улавливать сложные нелинейные взаимодействия между признаками, что может привести к ухудшению точности модели.

Пример графика логистической регрессии представлен на рисунке 1. Здесь синие точки – данные, относящиеся к разным классам, красная кривая, называемая сигмоидой – линия, разделяющая данные, принадлежащие разным классам, y – вероятность принадлежности объекта к классу, x – входные данные.

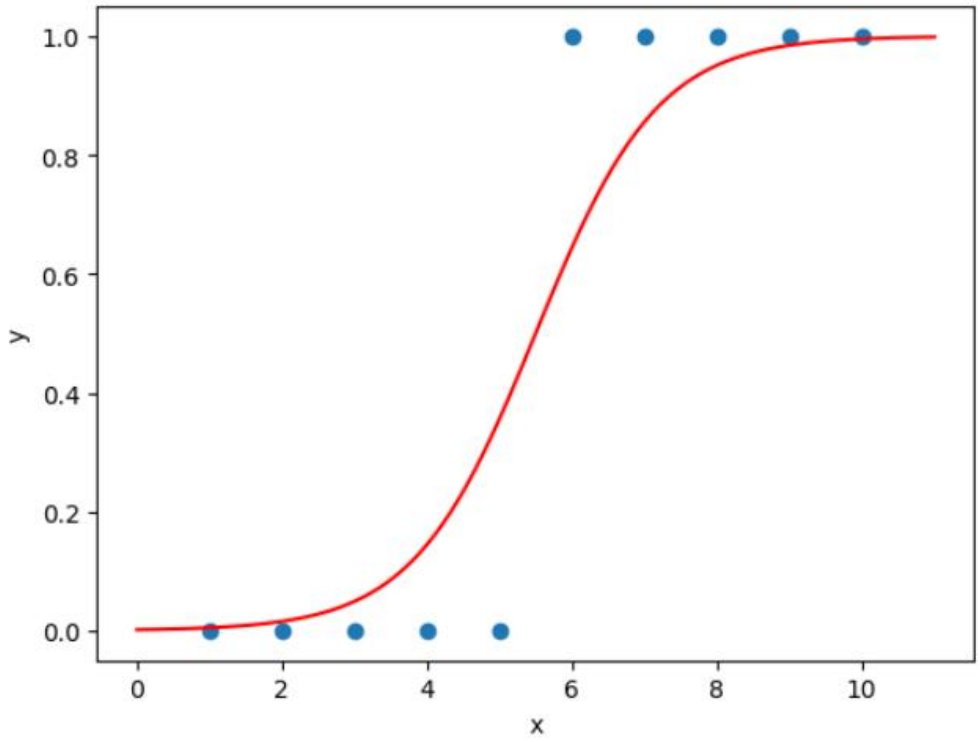

Рисунок 1 - График логистической регрессии

Метод опорных векторов (Support Vector Machines, SVM)

Метод опорных векторов - это модель машинного обучения, используемая для классификации [6]. Она работает путем построения гиперплоскости в n-мерном пространстве, которая разделяет данные на различные классы. Гиперплоскость определяется таким образом, чтобы максимизировать расстояние между ней и ближайшими объектами разных классов. Эти ближайшие объекты называются опорными векторами.

Преимуществами метода опорных векторов являются:

- модель может работать с данными в любом n-мерном пространстве;
- модель эффективно обрабатывает данные с большим количеством признаков.

Недостатком метода опорных векторов является вычислительная сложность данной модели.

Пример графика модели SVM представлен на рисунке 2. Здесь чёрная прямая – гиперплоскость, разделяющая данные на два класса, красные и синие точки – это данные, относящиеся к разным классам, x_1 и x_2 – входные данные.

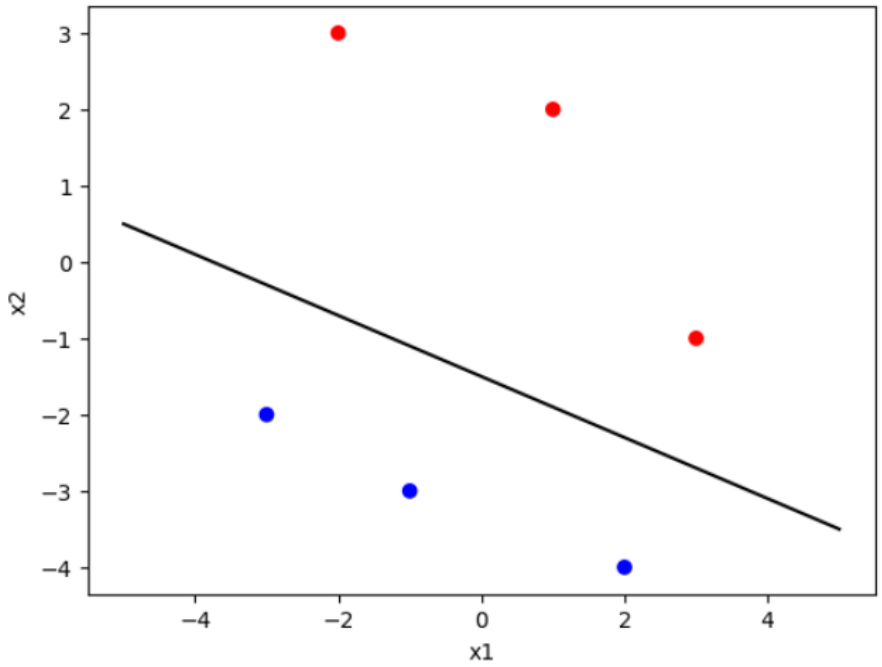

Рисунок 2 - График SVM

Наивный байесовский классификатор

Наивный байесовский классификатор — это модель машинного обучения, используемая для классификации данных [7]. Она основана на теореме Байеса, которая позволяет оценивать вероятность принадлежности объекта к определенному классу на основе априорных вероятностей и условных вероятностей признаков.

Для каждого класса определяется априорная вероятность, то есть вероятность того, что объект принадлежит данному классу, независимо от его признаков.

Для каждого класса определяются условные вероятности признаков, т.е. вероятности того, что объект принадлежит данному классу при заданных значениях признаков.

Для нового объекта вычисляются вероятности принадлежности к каждому классу на основе оценок априорных и условных вероятностей. Объект относится к тому классу, у которого наибольшая вероятность.

Преимуществом наивного байесовского классификатора является то, что для его обучения требуется меньше данных в сравнении с другими моделями.

Недостатком метода наивного байесовского классификатора является то, что предположение о независимости признаков может быть неверным в реальных данных, что приводит к потере точности.

Линейная регрессия

Линейная регрессия – это модель машинного обучения, которая используется для определения зависимости между независимыми переменными и зависимой переменной [8].

Принцип работы модели заключается в нахождении линейной функции, описывающей наилучшим образом зависимость между признаками и целевой переменной. Для этого используется метод наименьших квадратов, минимизирующий сумму квадратов отклонений предсказанных значений от реальных значений целевой переменной.

Преимуществом модели линейной регрессии является высокая скорость обучения.

Недостатком модели линейной регрессии является то, что модель линейной регрессии требует выполнения предположений о распределении данных и связи между переменными, что может ограничивать её применимость в некоторых ситуациях.

Пример графика линейной регрессии представлен на рисунке 3. Здесь красная прямая – линия, разделяющая данные на два класса, точки, находящиеся по разные стороны от прямой – данные, относящиеся к разным классам, x и y – входные данные.

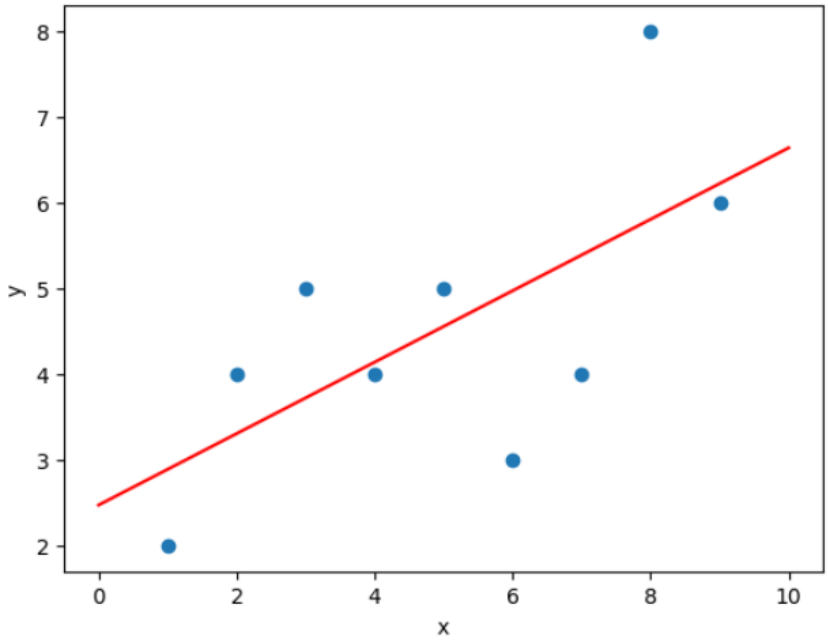

Рисунок 3 - График линейной регрессии

1.2.2. Глубокое обучение

Глубокое обучение - это раздел машинного обучения, использующий нейронные сети с несколькими слоями для извлечения высокоуровневых признаков из входных данных и предсказания целевых значений [10]. Глубокие нейронные сети состоят из множества слоев, каждый из которых выполняет определенные операции с данными. Эти слои могут быть свёрточными, рекуррентными или полносвязными, и каждый слой содержит набор параметров, которые настраиваются во время обучения.

Нейронные сети

Нейронные сети - это алгоритм машинного обучения, который используется для классификации, регрессии, обработки естественного языка и других задач [11].

Алгоритм нейронной сети:

1) создание модели нейронной сети: модель нейронной сети определяет архитектуру сети и параметры ее компонентов;

2) обучение нейронной сети: на данном этапе применяется выбранный алгоритм оптимизации, для обучения нейронной сети на обучающем наборе данных;

3) тестирование и оценка производительности: после обучения проверяется производительность нейронной сети на тестовом наборе данных.

Преимущества нейронных сетей:

- способность обрабатывать сложные данные;
- автоматическое извлечение признаков;
- устойчивость к выбросам;
- адаптивность к изменениям в данных.

Недостатки нейронных сетей:

- высокая вычислительная сложность;
- требуется много данных для обучения;
- неясность внутренней структуры;
- подверженность переобучению.

Пример нейронной сети представлен на рисунке 5. Здесь синие нейроны являются входными, красные нейроны являются скрытыми, зелёные нейроны являются выходными.

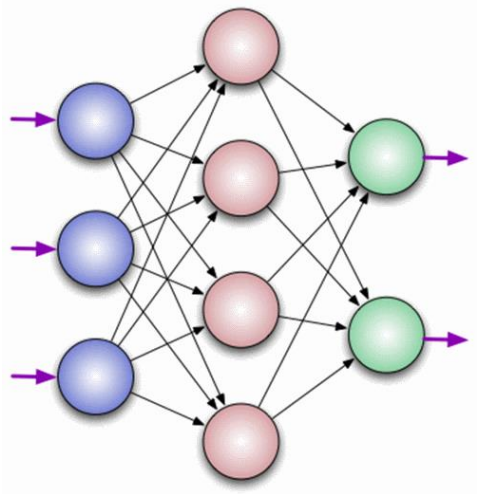

Рисунок 4 - Пример нейронной сети

1.2.3. Модели на основе решающих деревьев

Решающие деревья

Алгоритм решающих деревьев - это модель машинного обучения, используемая для решения задач классификации и регрессии [12]. Решающее

дерево представляет собой иерархическую структуру в виде дерева, в которой каждый узел представляет условие, а каждый листовой узел представляет класс или числовое значение для задач регрессии.

Алгоритм построения решающего дерева включает следующие шаги:

1) выбор признака, который будет использоваться для разделения данных на две подгруппы;

2) разделение данных на две подгруппы на основе значения выбранного признака;

3) повторение шагов 1-2 для каждой подгруппы, пока не будет достигнут критерий остановки;

4) присвоение классов или значений регрессии листовым узлам.

Преимущества модели решающих деревьев:

- легко интерпретировать, что позволяет анализировать причинно-следственные связи и принимать обоснованные решения;
- обучение решающего дерева происходит быстро и легко масштабируется для больших наборов данных;
- хорошо работают с большим количеством признаков;
- могут работать с данными разного типа, включая категориальные и числовые.

Недостатки модели решающих деревьев:

- могут быть склонны к переобучению;
- могут быть неустойчивы к небольшим изменениям в данных, что может привести к большим изменениям в структуре дерева;
- могут не справиться с задачами, где есть мультиколлинеарные зависимости между признаками;
- могут не всегда давать оптимальное решение для задачи, так как они выбирают локально оптимальное решение на каждом шаге построения дерева, а не глобально оптимальное.

Пример решающего дерева, классифицирующего исходную выборку на 5 классов, представлен на рисунке 5.

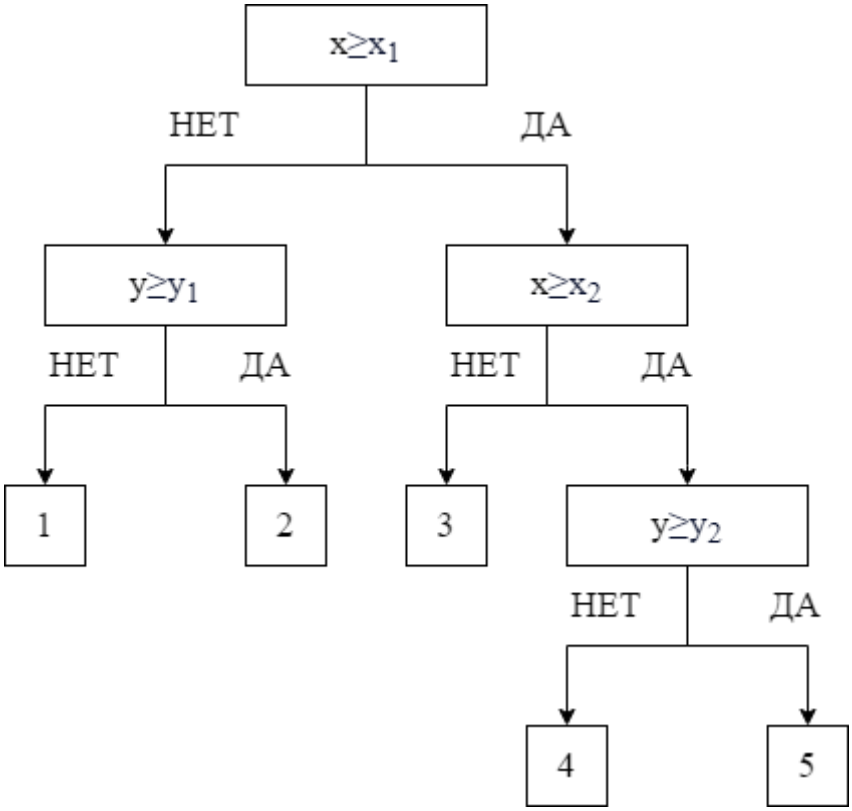

Рисунок 5 - Пример решающего дерева

Модель решающих деревьев является мощным инструментом для классификации и регрессии, который обладает рядом преимуществ перед другими методами машинного обучения. Решающие деревья легко интерпретируемы, позволяют обрабатывать данные с пропущенными значениями и выбросами, могут работать с категориальными признаками и могут быть использованы для анализа важности признаков. Кроме того, решающие деревья легко масштабируются, позволяют обрабатывать большие объемы данных и имеют быстрое время обучения и прогнозирования.

Случайный лес

Модель случайного леса — это алгоритм машинного обучения, который использует множество деревьев решений для классификации, регрессии и других задач [9]. Он является разновидностью ансамблевого метода обучения, в котором каждое дерево обучается на подмножестве данных и на подмножестве признаков.

Алгоритм работы модели случайного леса:

1) из обучающей выборки случайным образом выбирается подмножество размера n;

2) для этого подмножества строится дерево решений, но при каждом разбиении вместо всех признаков случайным образом выбирается подмножество размера m;

3) пункты 1 и 2 повторяются k раз для получения k деревьев решений;

4) каждое дерево решений прогнозирует класс объекта, итоговый результат определяется путем голосования среди всех деревьев;

5) важным параметром модели случайного леса является количество деревьев (k), которое должно быть выбрано для построения ансамбля. Также важно выбрать оптимальное значение параметра m, который определяет количество признаков, используемых при построении каждого дерева. Обычно в качестве значения m берется корень из общего числа признаков.

Преимущества модели случайного леса:

- обладает высокой точностью и способностью к обобщению на новых данных;
- может обрабатывать большие объемы данных и признаков.

Недостатки модели случайного леса:

- обучение может занимать много времени;
- при большом числе признаков, модель может потерять в точности, так как случайное подмножество признаков может не содержать наиболее информативные признаки.
- модель сложно интерпретировать, так как используется множество деревьев решений.

Пример модели случайного леса, состоящего из n деревьев представлен на рисунке 6.

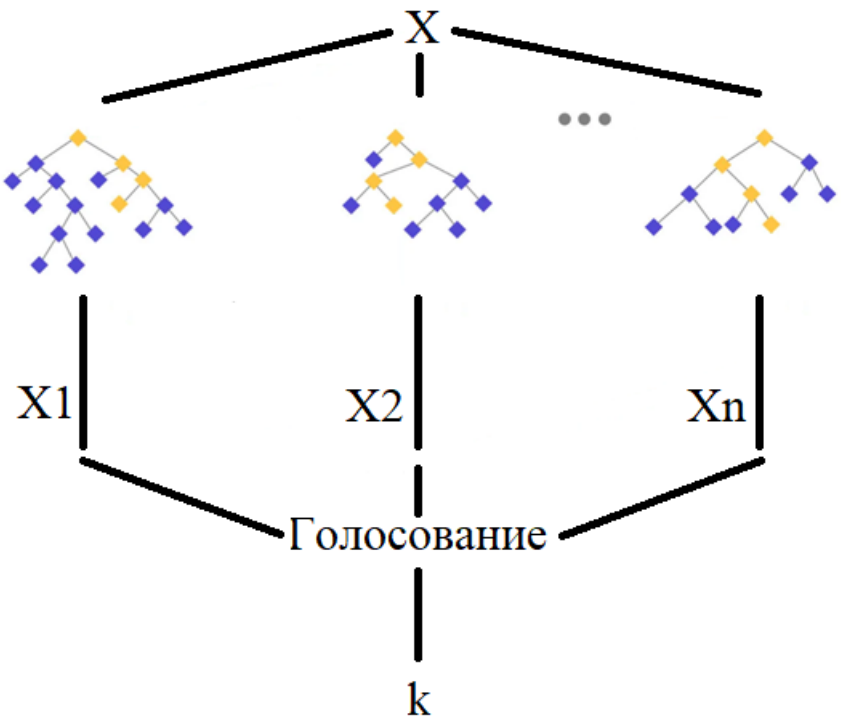

Рисунок 6 - Пример случайного леса

Бустинг

Модель бустинга решающих деревьев - это алгоритм машинного обучения, который основан на комбинации нескольких решающих деревьев для улучшения качества классификации или регрессии.

Алгоритм бустинга решающих деревьев включает следующие шаги:

1) обучение базового решающего дерева на обучающем наборе данных;

2) оценка ошибки базовой модели на обучающем наборе данных и создание весов для каждого элемента данных, учитывающих его сложность и вероятность правильной классификации;

3) создание нового решающего дерева, которое будет сфокусировано на ошибках предыдущей;

4) обновление весов элементов данных на основе ошибок предыдущей модели;

5) повторение шагов 2-4 для каждого последующего решающего дерева;

6) комбинация всех решающих деревьев в одну модель, используя взвешенное голосование для классификации или усреднение для регрессии.

Преимущества модели бустинга решающих деревьев:

- высокая точность вычислений;
- обработка больших объемов информации и способность работы с большим количеством признаков;
- не требует предварительной обработки данных;

Недостатки метода бустинга решающих деревьев:

- при использовании сложных моделей бустинга с большим числом деревьев, алгоритм может стать склонным к переобучению на тренировочных данных;
- выбросы в данных могут значительно повлиять на построение каждого нового дерева, так как алгоритм стремится минимизировать ошибку на всех данных;
- является вычислительно сложным алгоритмом.

1.3. Обзор методов интерпретации модели

Интерпретируемость моделей машинного обучения – это возможность кратко описать, почему модель работает (не вдаваясь в подробности).

Модели машинного обучения используются во многих сферах обеспечения жизни человека [13]. С каждым годом модели обрабатывают все больше данных и принимают все больше решений. Эти решения оказывают значимое влияние на людей.

Проблемой становится недоверие к полностью нечеловеческим моделям. Недоверие заключается в непонимании того, почему модели принимают то или иное решение и исходя из каких убеждений они действуют. Для решения проблемы недоверия стали применять методы интерпретации моделей.

Для решения задачи интерпретации моделей машинного обучения необходимо выбрать наиболее подходящие методы интерпретации для конкретных моделей и задач, а также провести эксперименты для оценки эффективности различных методов интерпретации. Важным аспектом является возможность объяснения решений, принятых моделью, что может помочь улучшить доверие к модели и ее использование в реальных приложениях.

Алгоритм для построения методов интерпретации моделей машинного обучения можно описать следующим образом:

1) выбор модели и ее обучение;

2) применение метода интерпретации;

3) визуализация результатов.

Методы интерпретации подразделяются на локальные и глобальные.

1.3.1. Локальные методы интерпретации

Локальные методы интерпретации относятся к методам, которые позволяют анализировать, какие признаки были наиболее важны для принятия решения моделью для конкретного объекта данных. Такие методы могут помочь в определении, почему модель приняла конкретное решение для данного объекта, и какие изменения в значениях признаков могут привести к изменению предсказания.

LIME (Local Interpretable Model-Agnostic Explanations)

LIME - это метод интерпретации моделей машинного обучения, который позволяет объяснять предсказания моделей на уровне отдельных объектов данных [14].

Суть метода LIME заключается в создании локальной модели, которая объясняет прогноз модели на конкретном объекте. Для этого метод LIME случайным образом создает подвыборку объектов, подобных заданному объекту, и генерирует для каждого из них "объясняющую модель", которая моделирует взаимодействие признаков для объяснения прогноза модели на этом объекте.

Затем LIME оценивает важность каждого признака для прогноза модели на данном объекте на основе вклада признака в объясняющую модель. В результате LIME выдает важность признаков для данного объекта, которые могут быть использованы для интерпретации прогноза модели.

Использование метода LIME позволяет получить интерпретируемые объяснения для каждого объекта данных, что может помочь понять, как модель работает на практике, и выявить возможные проблемы с моделью.

Пример интерпретации формулы $30 * x_1^2 + 50 * x_2^2 + 100 * x_3^2$ методом LIME представлен на рисунке 7. Здесь по оси абсцисс отложены значения вкладов признаков в обучающую модель, чем они больше по модулю, тем более значимым является признак, по оси ординат отложены сами признаки, для которых производится расчет.

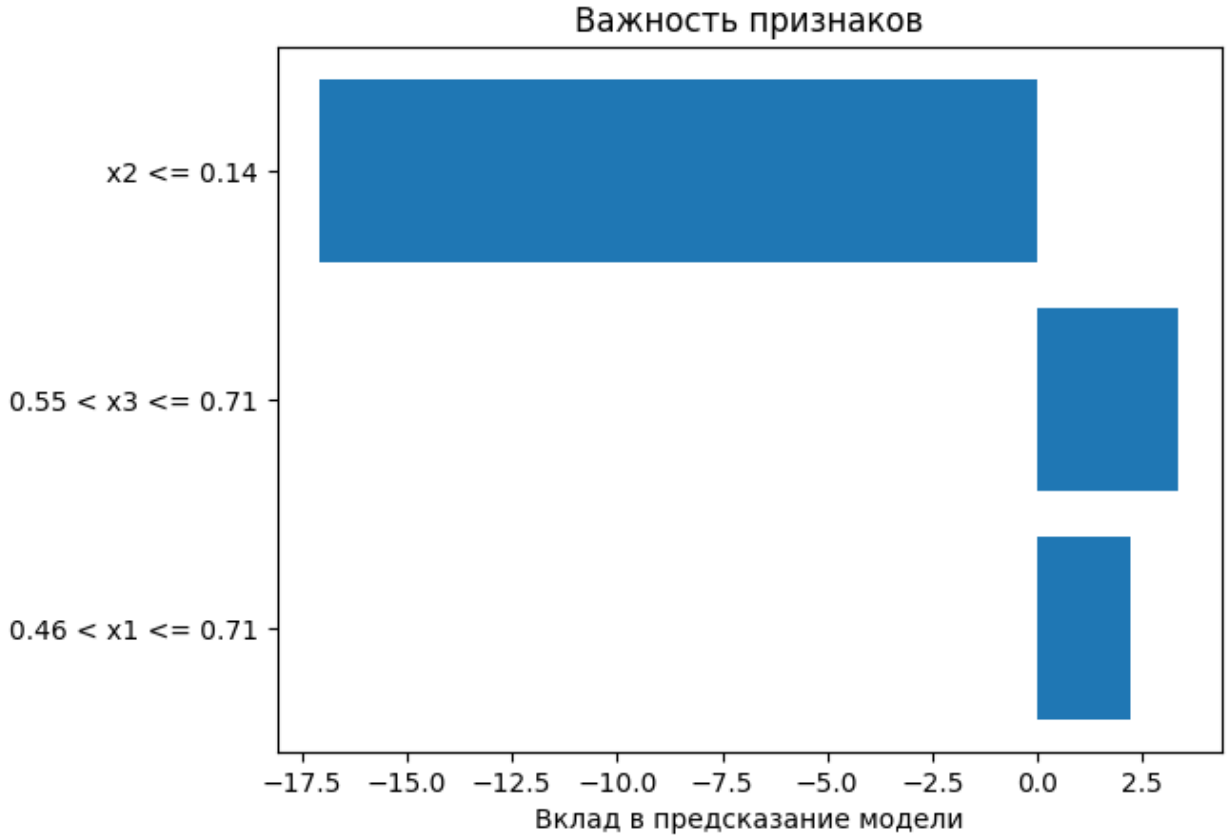

Рисунок 7 - Пример интерпретации методом LIME

ICE (Individual Conditional Expectation)

ICE - это метод интерпретации моделей машинного обучения, который позволяет оценить влияние каждого признака на прогноз модели на уровне отдельных объектов [15].

В основе метода ICE лежит идея разбиения диапазона значений каждого признака на интервалы и построения для каждого объекта графика, показывающего зависимость прогноза модели от значения данного признака в пределах его интервала. Таким образом, для каждого объекта строится свой график ICE, который позволяет оценить вклад каждого признака в прогноз модели для данного объекта.

Основное преимущество метода ICE заключается в его способности оценивать вклад каждого признака на уровне отдельных объектов, что позволяет получать более точные интерпретации прогнозов моделей машинного обучения. Кроме того, метод ICE не требует предположений о линейности зависимости между признаками и целевой переменной, что делает его универсальным для различных типов моделей машинного обучения.

Однако метод ICE также имеет свои недостатки, такие как сложность интерпретации графиков ICE для большого количества признаков и объектов, а также необходимость проведения дополнительных статистических тестов для оценки значимости различий между графиками ICE для разных объектов.

Пример интерпретации формулы $30 * x_1^2 + 50 * x_2^2 + 100 * x_3^2$ методом ICE представлен на рисунке 8. Здесь на оси абсцисс отложены значения признаков, на оси ординат отложены все значения прогноза модели для всех объектов.

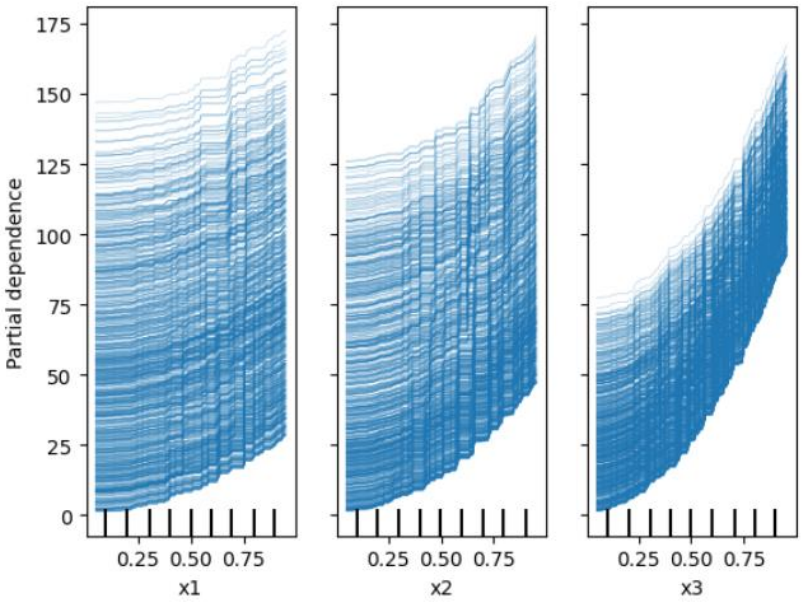

Рисунок 8 - Пример интерпретации методом ICE

SHAP (SHapley Additive exPlanations)

SHAP - это алгоритм, который позволяет интерпретировать причинно-следственные связи в моделях машинного обучения. Он определяет вклад каждого признака в предсказание модели [16].

Суть метода SHAP заключается в вычислении значений Шепли для каждого признака и объединении их в одну величину, которая показывает важность признака для прогноза модели на конкретном объекте. Для этого метод SHAP запускает модель машинного обучения несколько раз с разными комбинациями признаков и рассчитывает вклад каждого признака в прогноз на основе значений Шепли.

Метод SHAP также позволяет получать глобальные интерпретации модели, показывая, какие признаки наиболее важны для прогноза модели в целом. Кроме того, он может использоваться с различными типами моделей машинного обучения, включая линейные модели, деревья решений и нейронные сети.

TreeSHAP

TreeSHAP - это модификация метода SHAP, которая оптимизирует вычисления для деревьев решений.

Для вычисления значимости признаков метод TreeSHAP использует алгоритм, который проходит от корня до листьев дерева и вычисляет вклад каждого признака в каждое решающее правило на пути от корня до листа. Затем значения Шепли вычисляются путем агрегации вкладов признаков на всех возможных путях в дереве.

Основное преимущество метода TreeSHAP заключается в его способности рассчитывать значимость признаков для прогноза модели на уровне отдельных объектов. Это позволяет пользователям получать более точные и индивидуальные интерпретации прогнозов моделей, основанных на деревьях решений. Кроме того, метод TreeSHAP может использоваться с различными типами деревьев решений, включая случайный лес, градиентный бустинг и другие.

KernelSHAP

Метод KernelSHAP - это метод интерпретации моделей машинного обучения, основанный на методе Шепли значения и использующий ядерные методы для вычисления важности каждого признака в прогнозе модели.

Для вычисления значимости каждого признака метод KernelSHAP использует ядерную регрессию, которая позволяет оценить вклад каждого признака в прогноз модели на уровне отдельных объектов. В этом методе каждый объект рассматривается как "черный ящик", и значение прогноза модели для него рассчитывается на основе взвешенной суммы значений признаков для всех объектов. Затем используется ядерная регрессия для оценки вклада каждого признака в эту взвешенную сумму значений.

Пример интерпретации формулы $30 * x_1^2 + 50 * x_2^2 + 100 * x_3^2$ методом SHAP представлен на рисунке 9. Здесь по оси абсцисс отложены значения SHAP values, чем они выше, тем более значимым является значение признака, по оси ординат отложены сами признаки, для которых проведены расчёты, цвет точек отражает значение признака (чем краснее цвет, тем выше значение признака).

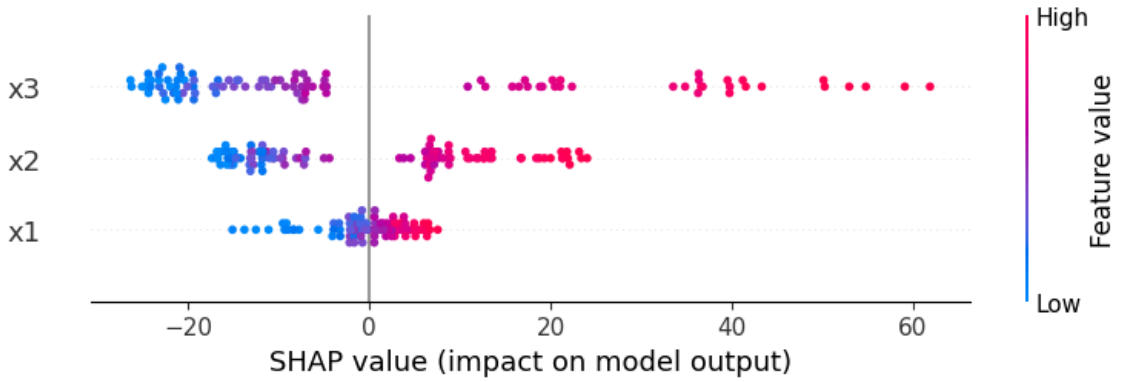

Рисунок 9 - Пример интерпретации методом SHAP

Метод SHAP представляет собой мощный инструмент для интерпретации прогнозов моделей машинного обучения, который имеет ряд преимуществ перед другими методами. SHAP способен объяснить, какие признаки оказывают наибольшее влияние на прогнозы модели и в какой степени, а также учитывать взаимодействия между признаками. Кроме того, SHAP гибок и может применяться к различным типам моделей, что позволяет использовать его для интерпретации прогнозов различных моделей машинного обучения. SHAP также обладает возможностью анализировать влияние группы признаков и влияние каждого признака на конкретный прогноз, что делает его полезным инструментом для анализа и улучшения моделей машинного обучения. Кроме

того, SHAP эффективен и может быть использован для анализа больших объемов данных, что делает его более оптимальным при выборе инструмента для решения задач машинного обучения в различных областях.

1.3.2. Глобальные методы интерпретации

Глобальные методы интерпретации относятся к методам, которые позволяют анализировать важность признаков в модели в целом. Они могут помочь в определении того, какие признаки являются наиболее важными для принятия решений моделью в целом и как они влияют на ее общую производительность.

PDP (Partial Dependence Plots)

Метод PDP - это метод интерпретации моделей машинного обучения, который позволяет оценить зависимость прогноза модели от отдельных признаков, учитывая влияние всех остальных признаков [15].

В основе метода PDP лежит идея оценки среднего значения прогноза модели для всех объектов, при фиксированных значениях определенного признака, варьируя значения всех остальных признаков.

Основное преимущество метода PDP заключается в том, что он позволяет выявлять нелинейные зависимости между признаками и целевой переменной.

Однако метод PDP также имеет свои недостатки, такие как ограниченность в выявлении сложных взаимодействий между признаками и необходимость предварительной обработки.

Пример интерпретации формулы $30 * x_1^2 + 50 * x_2^2 + 100 * x_3^2$ методом PDP представлен на рисунке 10. Здесь на оси абсцисс отложены значения признаков, на оси ординат отложены средние значения прогноза модели для всех объектов при соответствующих значениях всех остальных признаков (отмечены оранжевой пунктирной линией).

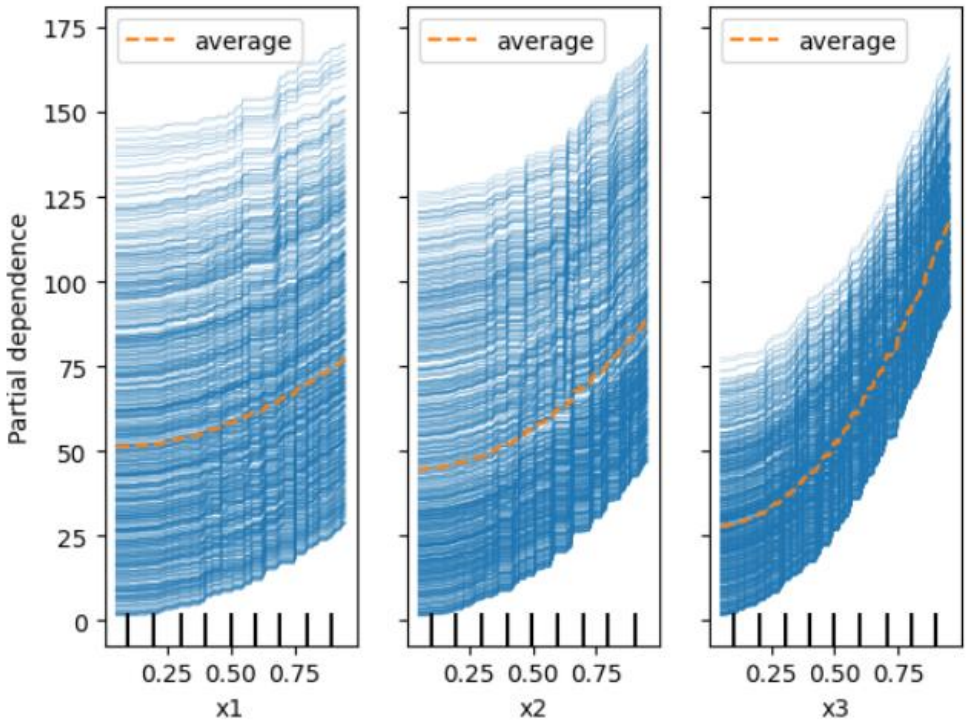

Рисунок 10 - Пример интерпретации методом PDP

Accumulated Local Effects (ALE) Plot

Accumulated Local Effects - это метод интерпретации моделей машинного обучения, который позволяет оценить влияние отдельных признаков на прогноз модели, учитывая все остальные признаки [17].

Основная идея метода ALE заключается в оценке локальных эффектов каждого признака на прогноз модели в различных точках диапазона значений этого признака. Для этого диапазон значений каждого признака разбивается на несколько интервалов, и в каждом интервале оценивается среднее значение изменения прогноза модели при изменении значения данного признака.

Затем полученные локальные эффекты суммируются по всем интервалам признака, чтобы получить накопленный эффект - общее влияние данного признака на прогноз модели при изменении его значений во всем диапазоне значений.

Основное преимущество метода ALE заключается в его способности оценивать влияние каждого признака на прогноз модели, учитывая все остальные признаки, что позволяет получать более точные и универсальные

интерпретации моделей машинного обучения. Кроме того, метод ALE позволяет выявлять нелинейные зависимости между признаками и целевой переменной.

Однако метод ALE также имеет свои недостатки, такие как потребность в предварительной обработке данных и оптимизации гиперпараметров модели для достижения наилучших результатов интерпретации.

1.4. Обзор ПО методов интерпретации

Существуют библиотеки и фреймворки, которые могут быть использованы для реализации методов интерпретации.

LIME

• *lime* - это библиотека для вычисления LIME значений, разработанная на *Python*. Она поддерживает большинство моделей машинного обучения, включая *Scikit-Learn, XGBoost, LightGBM, CatBoost* и другие. Она также имеет функции для визуализации и интерпретации LIME значений.

• *InterpretML* - это фреймворк машинного обучения, который обеспечивает широкий спектр функций для интерпретации моделей машинного обучения, включая вычисление LIME значений [18]. Он также поддерживает большинство моделей машинного обучения, включая *XGBoost, LightGBM, Scikit-Learn, TensorFlow* и другие.

ICE

• *pdpbox* - это библиотека для машинного обучения на *Python*, которая предоставляет инструменты для вычисления и визуализации ICE [19].

• *DALEX* - это открытый фреймворк для интерпретации моделей машинного обучения на *Python* и *R* [20]. *DALEX* включает функционал для вычисления ICE, который может быть использован вместе с моделями, созданными с помощью различных библиотек для машинного обучения.

• *FairML* - это библиотека на *Python* для интерпретации моделей машинного обучения, которая предоставляет инструменты для вычисления и визуализации ICE [21].

SHAP

• *shap* - это библиотека, написанная на языке *Python*, которая реализует метод SHAP для различных моделей машинного обучения [16]. Она включает функции для вычисления значимости признаков, глобальной значимости модели, а также графические инструменты для визуализации результатов.

• *XGBoost SHAP* - это библиотека, написанная на языке *Python*, которая реализует метод SHAP для моделей градиентного бустинга, созданных с помощью библиотеки *XGBoost* [22]. Она включает функции для вычисления значимости признаков, а также графические инструменты для визуализации результатов.

• *LightGBM SHAP* - это библиотека, написанная на языке *Python*, которая реализует метод SHAP для моделей градиентного бустинга, созданных с помощью библиотеки *LightGBM* [23]. Она включает функции для вычисления значимости признаков, а также графические инструменты для визуализации результатов.

• *CatBoost SHAP* - это библиотека, написанная на языке *Python*, которая реализует метод SHAP для моделей градиентного бустинга, созданных с помощью библиотеки *CatBoost* [24]. Она включает функции для вычисления значимости признаков, а также графические инструменты для визуализации результатов.

PDP

• *Scikit-learn* - это одна из самых популярных библиотек для машинного обучения на Python. Она также предоставляет функционал для вычисления и визуализации PDP [25].

• *PyCaret* - это открытый фреймворк для машинного обучения на *Python*. *PyCaret* включает функционал для вычисления и визуализации PDP, который может быть использован вместе с моделями, созданными в *PyCaret* [26].

• *H2O.ai* - это фреймворк для машинного обучения, который может быть использован на *Python* и других языках программирования [27]. *H2O.ai* включает

функционал для вычисления и визуализации PDP, который может быть использован с моделями, созданными в *H2O.ai*.

- *pdpbox*

ALE

- *PyALE* - это библиотека на *Python*, которая позволяет визуализировать ALE. Она имеет простой интерфейс и может быть использована для любой модели машинного обучения [28].

- *pdpbox*
- *DALEX*.
- scikit-learn

Интерпретация моделей машинного обучения является одной из наиболее развивающихся сфер машинного обучения. Для интерпретации разработано достаточно ПО, но не представлено ПО, автоматизирующее данный процесс. В рамках данной работы будет разработано и реализовано данное ПО.

2. Методика интерпретации моделей машинного обучения на основе решающих деревьев

2.1. Математическое описание метода решающих деревьев

Распознавание с помощью решающих деревьев

Предполагается, что бинарное дерево T используется для распознавания объектов, описываемых набором признаков $X_1, \ldots, X_n$. Каждой вершине v дерева T ставится в соответствие предикат, касающийся значения одного из признаков. Непрерывному признаку X_j соответствует предикат вида $X_j \geq \delta_v^j$, где δ_v^j - некоторый пороговый параметр. Категориальному признаку $X_{j'}$, принимающему значения из множества $M_{j'} = \{a_1^{j'}\}$ ставится в соответствие предикат вида $X_{j'} \in M_{j'}^{v_1}$ " , где $M_{j'}^{v_1}$ является элементом дихотомического разбиения $\left\{M_{j'}^{v_1}, M_{j'}^{v_2}\right\}$ множества $M_{j'}$. Выбор одного из двух, выходящих из вершины v рёбер производится в зависимости от значения предиката.

Процесс распознавания заканчивается при достижении концевой вершины (листа). Объект относится классу согласно метке, поставленной в соответствие данному листу.

Обучение решающих деревьев

Рассматривается задача распознавания с классами $K1, \ldots, K_L$. Обучение производится по обучающей выборке S_t и включает в себя поиск оптимальных пороговых параметров или оптимальных дихотомических разбиений для признаков $X_1, \ldots, X_n$. При этом поиск производится исходя из требования снижения среднего индекса неоднородности в выборках, порождаемых искомым дихотомическим разбиением обучающей выборки S_t.

Индекс неоднородности вычисляется для произвольной выборки S_t, содержащей объекты из классов $K_1, \ldots, K_L$. При этом используется несколько видов индексов, включая: энтропийный индекс неоднородности, индекс Джини, индекс ошибочной классификации.

Энтропийный индекс неоднородности вычисляется по формуле

$$\gamma_e(S) = -\sum_{i=1}^{L} P_i ln P_i,$$

где P_i - доля объектов класса K_i в выборке S. При этом принимается, что $0\, ln(0) = 0$. Наибольшее значение $\gamma_e(S)$ принимает при равенстве долей классов. Наименьшее значение $\gamma_e(S)$ достигается при принадлежности всех объектов одному классу.

Индекс Джини вычисляется по формуле

$$\gamma_g(S) = 1 - \sum_{i=1}^{L} P_i^2.$$

Индекс ошибочной классификации вычисляется по формуле

$$\gamma_m(S) = 1 - \max_{1,\dots,\mathrm{L}} (P_i).$$

Можно сделать вывод, что индексы ошибочной классификации и Джинни также достигают минимального значения при принадлежности всех объектов обучающей выборке одному классу.

Предположим, что в методе обучения используется индекс неоднородности γ_*. Для оценки эффективности разбиения обучающей выборки S_t на непересекающиеся подвыборки S_t^l и S_t^r используется уменьшение среднего индекса неоднородности в S_t^l и S_t^r по отношению к S_t. Данное уменьшение вычисляется по формуле

$$\Delta(\gamma_*, S_t) = \gamma_*(S_t) - P_{l\gamma_*}(S_t^l) - P_{r\gamma_*}(S_t^r)),$$

где P_l и P_r являются долями S_t^l и S_t^r в выборке S_t.

На первом этапе обучения бинарного решающего дерева ищется оптимальный предикат соответствующий корневой вершине. С этой целью оптимальные разбиения строятся для каждого из признаков из набора $X_1, \dots, X_n$. Выбирается признак $X_{i_{max}}$ с максимальным значением индекса $\Delta(\gamma_*, S_t)$. Подвыбороки S_t^l и S_t^r, задаваемые оптимальным предикатом для $X_{i_{min}}$ оцениваются с помощью критерия остановки.

Критерии остановки

В качестве критерия остановки может быть использован простейший критерий достижения полной однородности по одному из классов. В случае, если какая-та из выборок S_t^* удовлетворяет критерию остановки, то соответствующая вершина дерева объявляется концевой и для неё вычисляется метка класса. В случае, если выборка S_t^* не удовлетворяет критерию остановки, то формируется новая внутренняя вершина, для которой процесс построения дерева продолжается. Однако вместо обучающей выборки S_t используется соответствующая вновь образованной внутренней вершине v выборка S_v, которая равна S_t^*. Для данной выборки производятся те же самые построения, которые на начальном этапе проводились для обучающей выборки S_t. Обучение может проводиться до тех пор, пока все вновь построенные вершины не окажутся однородными по классам. Такое дерево может быть построено всегда, когда обучающая выборка не содержит объектов с одним и тем же значениям каждого из признаков, принадлежащих разным классам. Однако абсолютная точность на обучающей выборке не всегда приводить к высокой обобщающей способности в результате эффекта переобучения. Одним из способов достижения более высокой обобщающей способности является использования критериев остановки, позволяющих остановит процесс построения дерева до того, как будет достигнута полная однородность концевых вершин.

Примеры критериев остановки:

• ограничение максимальной глубины дерева;

• ограничение минимального числа объектов в листе;

• ограничение максимального количества листьев в дереве;

• остановка в случае, если все объекты в листе относятся к одному классу.

Стрижка дерева

Использование критериев остановки не всегда позволяет адекватно оценить необходимую глубину дерева. Слишком ранняя остановка ветвления может привести к потере информативных предикатов, которые могут быть на самом деле найдены только при достаточно большой глубине ветвления.

В связи с этим нередко целесообразным оказывается построение сначала полного дерева, которое затем уменьшается до оптимального с точки зрения достижения максимальной обучающей способности размера путём объединения некоторых концевых вершин. Такой процесс в литературе принято называть «pruning» («стрижка»). При стрижке дерева может быть использован критерий целесообразности объединения двух вершин, основанный на сравнение на контрольной выборке точности распознавания до и после проведения «стрижки». Ещё один способ оптимизации обобщающей способности деревьев основан на учёте при «стрижке» дерева до некоторой внутренней вершины v одновременно увеличения точности разделения классов на обучающей выборке и увеличения сложности, которые возникают благодаря ветвлению из v.

При этом прирост сложности, связанный с ветвлением из вершины v, может быть оценён через число листьев в поддереве T_v^{sub} полного решающего дерева с корневой вершиной v. Следует отметить, что рост сложности является штрафующим фактором, компенсирующим прирост точности разделения на обучающей выборке с помощью включения поддерева T_v^{sub} в решающее дерево. Разработан целый ряд эвристических критериев, которые позволяют оценить целесообразность включения T_v^{sub}. Данные критерии учитывают одновременно сложность и разделяющую способность.

2.2. Математическое описание метода LIME

Математически метод LIME обосновывается следующим образом:

$$E(x) = arg \min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Модель объяснения для объекта x – это локальная модель g, которая минимизирует функцию потерь L, которая измеряет, насколько близким является объяснение к прогнозу исходной модели машинного обучения f, при этом сохраняя низкую сложность модели $\Omega(g)$. G - это семейство возможных объяснений. Мера близости π_x определяет размер окрестности вокруг объекта x, которая рассматривается для объяснения.

2.3. Математическое описание метода PDP

Частичная функция зависимости для задачи регрессии определяется следующим образом:

$$f_S(x_s) = E_{X_C}[f(x_S, X_C)] = \int f(x_S, X_C) dP(X_C)$$

Здесь x_S представляют собой признаки, для которых должна быть построена частичная функция зависимости, X_C - это остальные признаки, используемые в модели машинного обучения f, которые здесь рассматриваются как случайные переменные. Обычно в наборе S присутствует только один или два признака. Признаки в S - это те признаки, для которых необходимо произвести интерпретацию. Векторы признаков x_S и X_C в совокупности составляют полное пространство признаков x. Частичная зависимость работает путем маргинализации результатов работы модели машинного обучения по распределению признаков из набора C, так что функция показывает связь между интересующими признаками из набора S и прогнозируемым результатом. Путем маргинализации по остальным признакам получается функция, зависящая только от признаков из S, с учетом взаимодействий с другими признаками.

Частичная функция f_S оценивается путем расчета средних значений на тренировочных входных данных, также известный как метод Монте-Карло:

$$f_S(x_S) = \frac{1}{n} \sum_{i=1}^{n} f(x_S, x_C^{(i)})$$

Частичная функция позволяет определить средний маргинальный эффект на предсказание при заданных значениях признаков S. В этой формуле $x_C^{(i)}$ представляют собой значения признаков из набора данных для признаков, которые не рассматриваются, n - количество объектов в наборе данных. Предположение PDP состоит в том, что признаки в C не коррелируют с признаками из S. Если это предположение нарушается, то средние значения, рассчитанные для графика частичной зависимости, могут включать точки данных, которые маловероятны или даже невозможны.

Для классификации график частичной зависимости отображает вероятность принадлежности к классу при различных значениях признаков S. Для каждой из категорий мы получаем оценку PDP, задавая всем объектам одну категорию.

Для числовых признаков важность определяется отклонением каждого уникального значения признака от средней кривой:

$$I(x_S) = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}\left(f_S\left(x_S^{(k)}\right) - \frac{1}{K}\sum_{k=1}^{K} f_S(x_S^{(k)})\right)^2},$$

здесь $x_S^{(k)}$ представляет собой K уникальных значений признака X_S.

2.4. Математическое описание метода SHAP

В основе метода SHAP используется концепция теории кооперативных игр, известная как значимость Шепли. Значимость Шепли - это метод распределения выгоды между участниками коалиции, основанный на их вкладе в коалицию.

2.4.1. Значения Шепли в теории игр

Кооперативной игрой называется игра, в которой коалиция игроков действует совместно. С середины XX века известны так называемые значения Шепли или Shapley values, которые позволяют численно оценить вклад каждого игрока в достижение общей цели.

Определение Shapley values

Пусть существует характеристическая функция v, которая каждому множеству игроков сопоставляет число - эффективность данной коалиции игроков. Тогда Shapley value для каждого игрока - это число, рассчитываемое по формуле (19). Обозначим за $\Delta(i, s)$ прирост эффективности от добавления игрока i в коалицию игроков S:

$$\Delta(i, s) = (S \cup i) - v(S), \quad (19)$$

Пусть всего есть N игроков. Рассмотрим множество П всех возможных упорядочиваний игроков, и обозначим за p множество игроков, стоящих перед

игроков i в упорядочивании π. Shapley value для игрока i рассчитывается таким образом:

$$\phi(i) = \frac{1}{N!} \sum_{\pi \in \Pi} \Delta\big(i, (p)\big). \quad (20)$$

То есть считается средний прирост эффективности от добавления i-го игрока в коалицию игроков, стоящих перед ним, по всем возможным упорядочиваниям игроков.

Формула (20) задается аксиоматически. Так как $\Delta(i, S)$ не зависит от порядка игроков в S, то можно объединить равные друг другу слагаемые и переписать формулу (20) в следующем эквивалентном виде:

$$\phi(i) = \sum_{S \subseteq \{1,2,\dots,N\} \setminus i} \frac{|S|!\,(|N| - |S| - 1)!}{N!} \Delta(i, S), \quad (21)$$

Формула (21) является взвешенной суммой по всем подмножествам игроков, не содержащих игрока i, в которой веса принимают наибольшие значения при $|S| \approx 0$ или $|S| \approx |N|$ и наименьшие значения при $|S| \approx \frac{|N|}{2}$.

2.4.2. Регрессионные значения Шепли

Shapley values можно применить в машинном обучении, если игроками считать наличие отдельных признаков, а результатом игры - ответ модели на конкретном примере x.

Регрессионные значения Шепли или Shapley regression values позволяют оценить вклад каждого признака в ответ модели f. Зафиксируем конкретный тестовый пример x и обучающую выборку, за характеристическую функцию множества признаков возьмём предсказание модели, обученной только на этих признаках:

$$v(S) = f_S(x_S). \quad (22)$$

Тогда $\Delta(i, S)$ - изменение в предсказании x между моделью $f_{S \cup \{i\}}$, обученной на признаках $S \cup \{i\}$ и моделью f_S, обученной на признаках S будет равняться:

$$\Delta(i,S) = \left(f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S(x_S)\right). \quad (23)$$

Тогда вклад отдельных признаков в величину предсказания модели можно оценивать по формулам (20) и (21). Стоит отметить, что рассматривается не вклад каждого признака в точность модели, а вклад каждого признака в величину предсказания модели на конкретном тестовом примере, что помогает интерпретировать это предсказание.

В Shapley regression values сравнивается текущее значение признака на примере x с его полным отсутствием при обучении и тестировании.

2.4.3. Значения SHAP

Для аппроксимации Shapley regression values на одной обучающей модели на всех признаках необходимо получать предсказание модели в случаях, когда многие из признаков имеют неопределенные значения.

Применим статистический подход и будем считать, что обучающие и тестовые данные взяты из некоторого распределения вероятностей. Пусть часть признаков в примере x известны, часть пропущены. За x_S обозначаются известные признаки. В SHAP характеристическая функция множества признаков S для примера x и модели f задается как условное математическое ожидание: $v(S) = E[f(x)|x_S]$. Данная формула означает, что за $v(S)$ берется математическое ожидание предсказания f на примерах x', взятых из распределения данных, таких, что $x'_S = x_S$.

Определение значения SHAP

Пусть имеется модель f, распределение данных и тестовый пример x. Необходимо оценить важность текущих значений каждого признака по сравнению с их неопределенными значениями. Значения SHAP или SHAP values для признаков на примере x - это Shapley values, рассчитываемые для следующей кооперативной игры:

- Игроками являются признаки (наличие i-го игрока означает текущее значение i-го признака на примере x, отсутствие i-го игрока означает неопределенное значение i-го признака - так же, как в Shapley regression values).

• Характеристической функцией $v(S)$ коалиции признаков S является условное математическое ожидание $E[f(x)|x_S]$ по распределению данных.

Таким образом, алгоритм расчета SHAP values следует формулам (20) и (21): для каждого возможного упорядочивания признаков берутся все признаки, стоящие перед i-м признаком (обозначаются за S) и считается величина

$$\Delta_f(i,S) = E[f(x)|x_{S\cup i}] - E[f(x)|x_S], \quad (24)$$

после чего полученные значения усредняются по всем упорядочиваниям. Это означает, что SHAP values описывают ожидаемый прирост выходного значения модели при добавлении i-го признака в текущем примере.

Отличие SHAP values от Shapley regression values в том, что в последних характеристической функцией группы признаков x_S является значение $f_{x_s}(x_S)$, а в SHAP values - $E[f(x)|x_S]$. В целом эти значения близки, так как f_{x_S} как правило моделирует $E[y|x_S]$. Но Shapley regression values требуют многократного обучения модели и таким образом являются характеристикой обучаемой модели, тогда как SHAP values являются характеристикой обученной модели.

3. Программная часть

3.1. Программная реализация

Для реализации ПО, автоматизирующего интерпретацию моделей машинного обучения, был выбраны язык программирования *Python* с множеством библиотек, по причине удобства использования для поставленной задачи.

Библиотека *SHAP*

SHAP - это библиотека машинного обучения, предназначенная для объяснения предсказаний моделей. Она основана на теории кооперативных игр и использует концепцию значения Шепли для определения вклада каждого признака в предсказание модели.

SHAP предоставляет методы для вычисления важности признаков и объяснений для каждого отдельного предсказания. Библиотека может использоваться с различными типами моделей машинного обучения, включая линейные модели, деревья решений, нейронные сети и многие другие.

Одной из ключевых особенностей *SHAP* является его способность предоставлять объяснения, которые удовлетворяют свойству справедливости и консистентности. Это означает, что вклад каждого признака в объяснение суммируется, чтобы соответствовать фактическому предсказанию модели.

Библиотека *LIME*

LIME - это библиотека машинного обучения, предназначенная для объяснения предсказаний моделей машинного обучения. Она разработана для обеспечения прозрачности и интерпретируемости моделей, которые в противном случае могут быть сложными для понимания.

Основная идея *LIME* заключается в создании локальных интерпретируемых моделей для объяснения предсказаний. Она работает путем генерации интерпретируемых "объяснителей" для отдельных предсказаний модели. Библиотека *LIME* предлагает методы для создания объяснителей, которые используют линейные модели, такие как логистическая регрессия, для

приближения поведения исходной модели в окрестности конкретного предсказания.

LIME может использоваться с различными моделями машинного обучения и типами данных, включая текстовые данные, изображения и табличные данные. Библиотека предоставляет удобные инструменты для создания объяснений, визуализации результатов и оценки важности признаков для предсказаний модели.

Библиотека *XGBoost*

Библиотека *XGBoost* является эффективной библиотекой с открытым исходным кодом, специализирующейся на реализации алгоритма градиентного бустинга над решающими деревьями. Библиотека отличается высокой скоростью работы и обладает высокой производительностью. Поэтому *XGBoost* пользуется популярностью при решении задач машинного обучения с использованием табличных наборов данных.

Библиотека *Scikit-learn*

Scikit-learn является одним из наиболее распространенных и популярных пакетов *Python* для машинного обучения. Он предоставляет обширный набор функций и алгоритмов, которые позволяют выполнить различные операции в области анализа данных. *Scikit-learn* обладает широким спектром возможностей, включая предварительную обработку данных, уменьшение размерности, выбор модели для регрессии или классификации.

Библиотека *Pandas*

Pandas - это библиотека на языке Python, предназначенная для обработки и анализа данных. Она предоставляет мощные инструменты для работы с данными в среде *Python*, обеспечивая не только сбор и очистку данных, но также их анализ и моделирование без необходимости переключения на специализированные языки для статистической обработки данных. Основное предназначение *Pandas* - это облегчить очистку и первичную оценку данных с помощью широкого спектра операций.

Библиотека Matplotlib

Matplotlib - это библиотека на языке программирования *Python*, предназначенная для визуализации. Она предоставляет широкий спектр возможностей для создания различных видов графиков и диаграмм, включая линейные графики, диаграммы рассеяния, столбчатые и круговые диаграммы, диаграммы стебель-листья, контурные графики, поля градиентов и спектральные диаграммы.

Программная реализация ПО, автоматизирующего интерпретацию моделей машинного обучения, включает в себя:

1) создание и обучение модели машинного обучения, которую необходимо интерпретировать;
2) интерпретация модели машинного обучения различными методами;
3) визуализация интерпретации модели машинного обучения.

Для реализации ПО создан класс *Inter*, включающий в себя методы класса, реализующие поставленные задачи. В качестве входных параметров класс *Inter* принимает набор данных *data*, целевую переменную *metka*, номер объекта *idd*, который необходимо интерпретировать локально, тип решаемой задачи *model_type*. Метод класса *__init__* производит инициализацию входных параметров. Метод класса *model* создает и обучает модель машинного обучения с помощью метода градиентного бустинга над решающими деревьями. Метод класса *inter_global* глобально интерпретирует модель машинного обучения методами SHAP и PDP и затем визуализирует результаты интерпретации. Метод класса *inter_local* локально интерпретирует модель машинного обучения методом LIME и затем визуализирует результаты интерпретации. Блок-схема класса, описывающая его работу, представлена на рисунке 11.

Рисунок 11 - Блок-схема класса *Inter*

3.2. Вычислительные эксперименты

В рамках данной работы вычислительный эксперимент подразумевает под собой интерпретацию моделей машинного обучения с помощью созданного класса *Inter*.

3.2.1. Эксперимент 1

В первом эксперименте продемонстрированы результаты интерпретации методов машинного обучения на примере набора данных *diabetes*, предоставляемого Университетом Джонса Хопкинса. Он состоит из объектов, каждый из которых имеет восемь признаков и одну целевую переменную, количество объектов равняется 768.

Признаками объектов являются:

- Pregnancies - количество беременностей;
- Glucose - плазменные концентрации глюкозы в крови;
- BloodPressure - диастолическое артериальное давление;
- SkinThickness - толщина кожи в области трицепса;
- Insulin - количество инсулина в крови;
- BMI - индекс массы тела;
- DiabetesPedigreeFunction - оценка предрасположенности к диабету;
- Age - возраст.

Целевой переменной является метка класса: 0 - прогноз, на не предрасположенность к заболеванию сахарным диабетом в ближайшие пять лет, 1 — прогноз, на предрасположенность к заболеванию сахарным диабетом в ближайшие пять лет.

Целью проведения данного эксперимента является демонстрация работы ПО для автоматизации интерпретации методов машинного обучения для задачи классификации.

Результаты интерпретации глобальными методами представлены на рисунках 12- 14. Результаты интерпретации локальным методом представлены на рисунках 15 и 16.

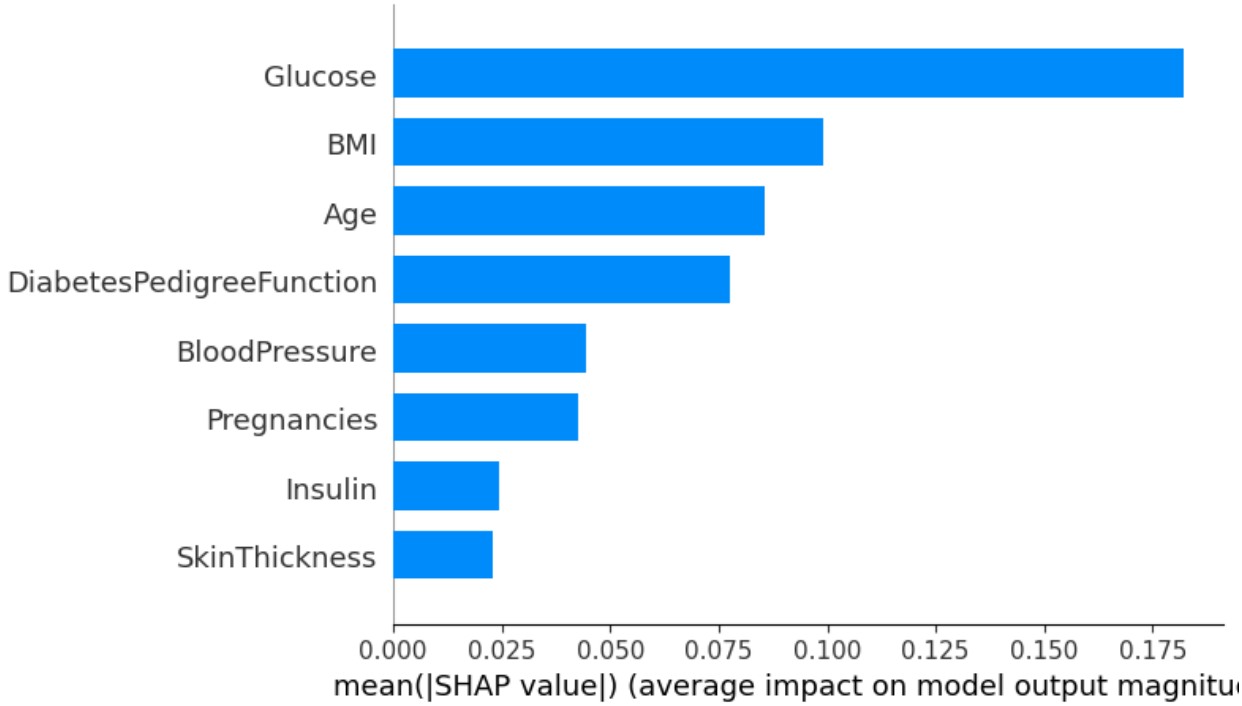

Рисунок 12 - Глобальная интерпретация модели машинного обучения, построенной на основе набора данных *diabetes*, методом SHAP

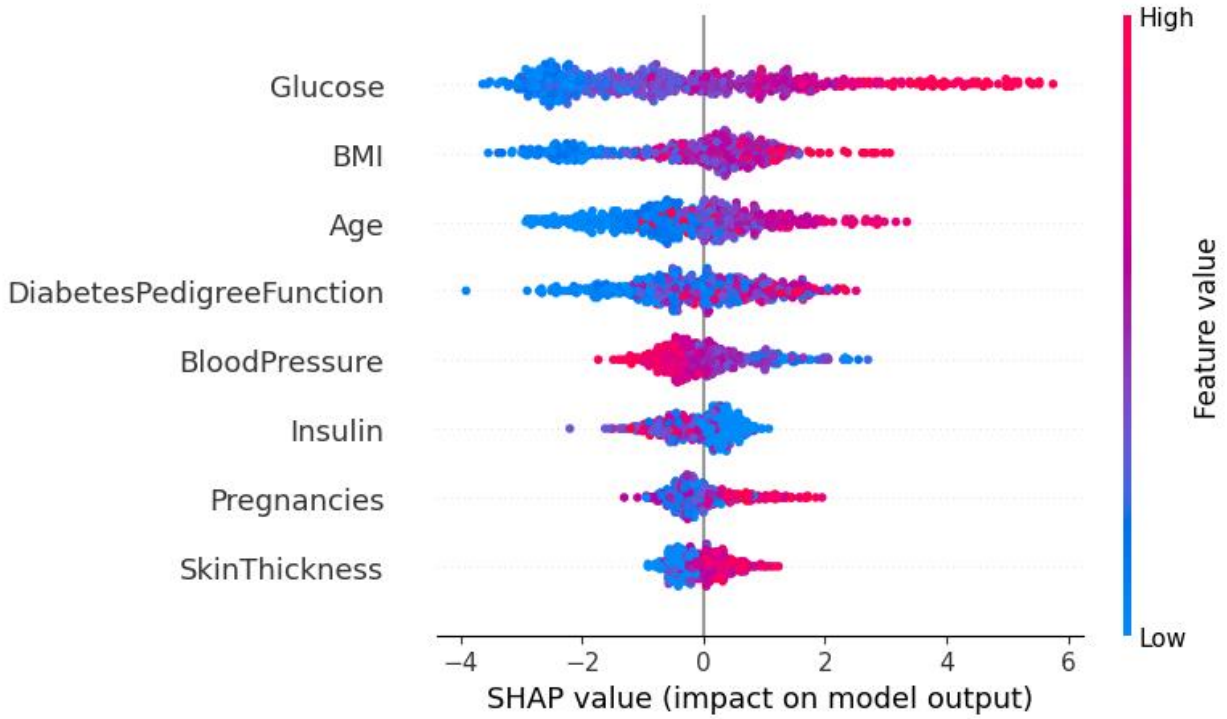

Рисунок 13 - Глобальная интерпретация модели машинного обучения, построенной на основе набора данных *diabetes*, методом SHAP

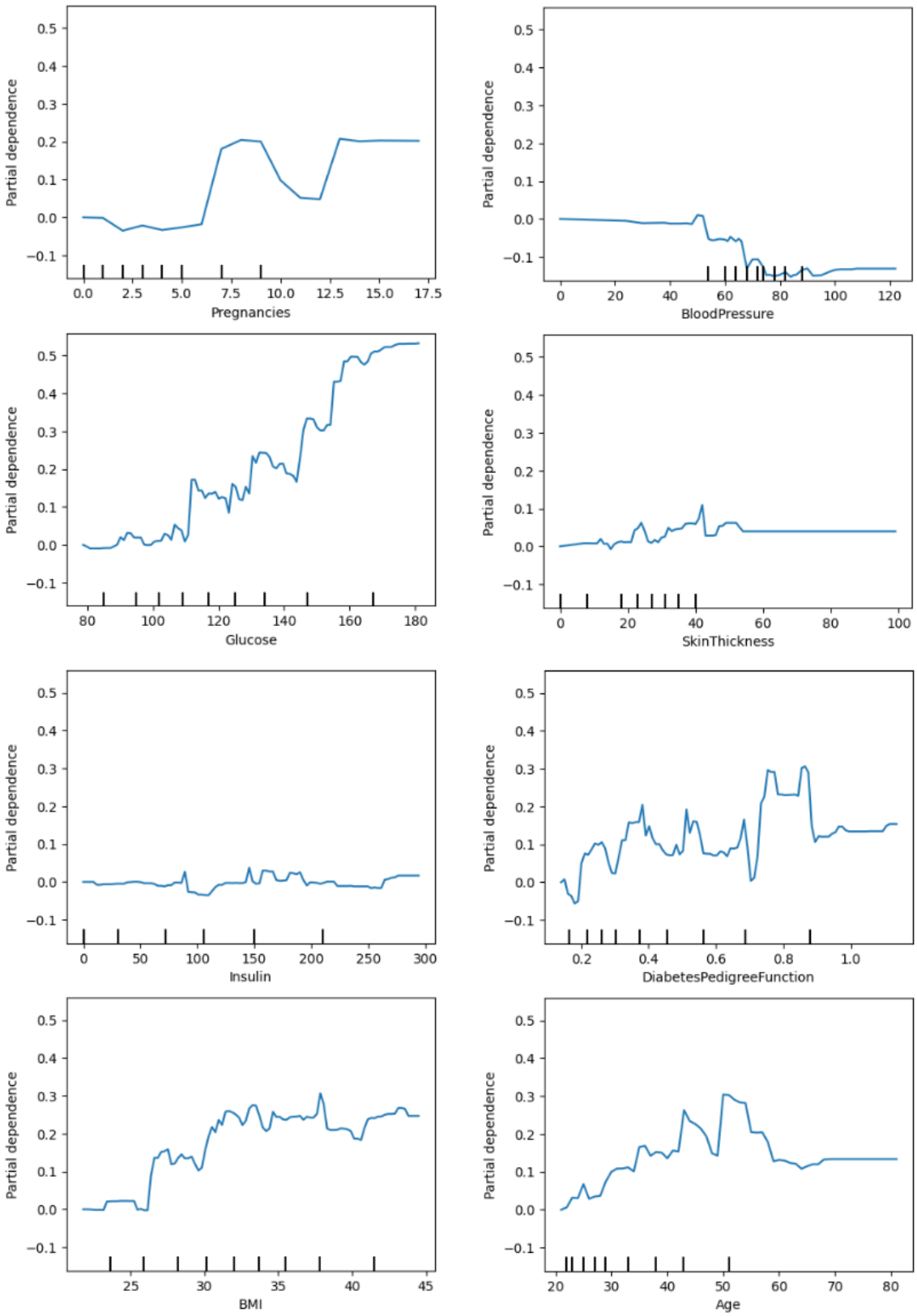

Рисунок 14 - Глобальная интерпретация модели машинного обучения, построенной на основе набора данных *diabetes*, методом PDP

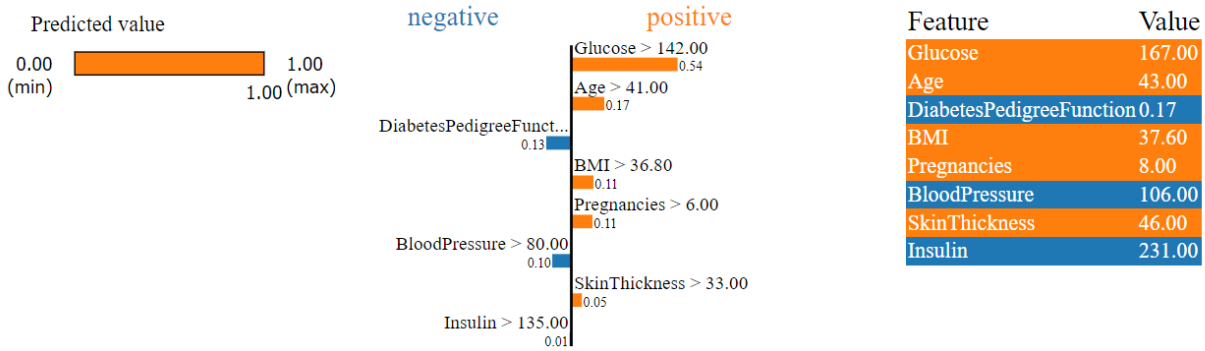

Рисунок 15 - Локальная интерпретация модели машинного обучения для объекта №60, построенной на основе набора данных *diabetes*, методом LIME

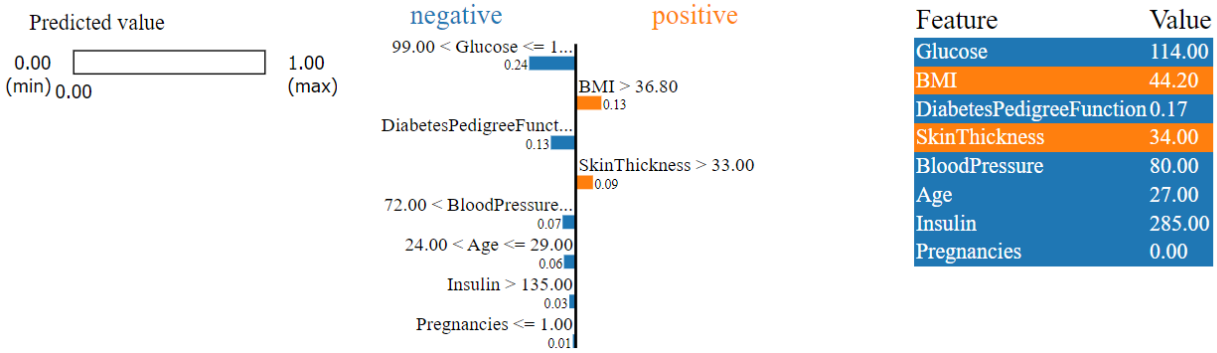

Рисунок 16. Локальная интерпретация модели машинного обучения для объекта №100, построенной на основе набора данных *diabetes*, методом LIME.

3.2.2. Эксперимент 2

Во втором эксперименте продемонстрированы результаты интерпретации методов машинного обучения на примере набора данных *california housing*, который содержит данные о средней стоимость домов в Калифорнии в зависимости от квартала. Набор данных состоит из объектов, каждый из которых имеет восемь признаков и одну целевую переменную, количество объектов равняется 20640.

Признаками объектов являются:

- Longitude - долгота квартала с недвижимостью;
- Latitude - широта квартала с недвижимостью;
- HouseAge - медиана возраста домов в квартале;
- AveRooms - общее количество комнат в квартале;
- AveBedrms - общее количество спален в квартале;

- Population - население квартала;
- AveOccup - количество семей в квартале;
- MedInc - медианный доход в квартале.

Целевой переменной является медианная стоимость дома в квартале.

Целью проведения данного эксперимента является демонстрация работы ПО для автоматизации интерпретации методов машинного обучения для задачи регрессии.

Результаты интерпретации глобальными методами представлены на рисунках 17- 19. Результаты интерпретации локальным методом представлены на рисунках 20 и 21.

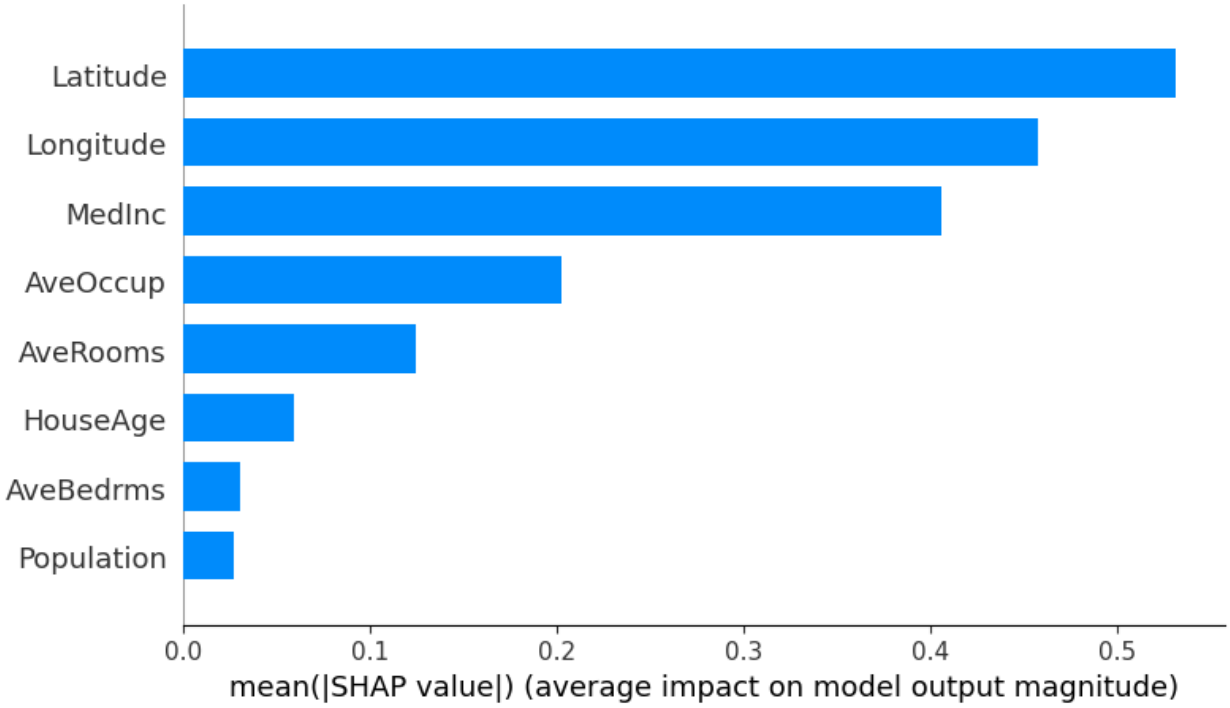

Рисунок 17 - Глобальная интерпретация модели машинного обучения, построенной на основе набора данных california housing, методом SHAP

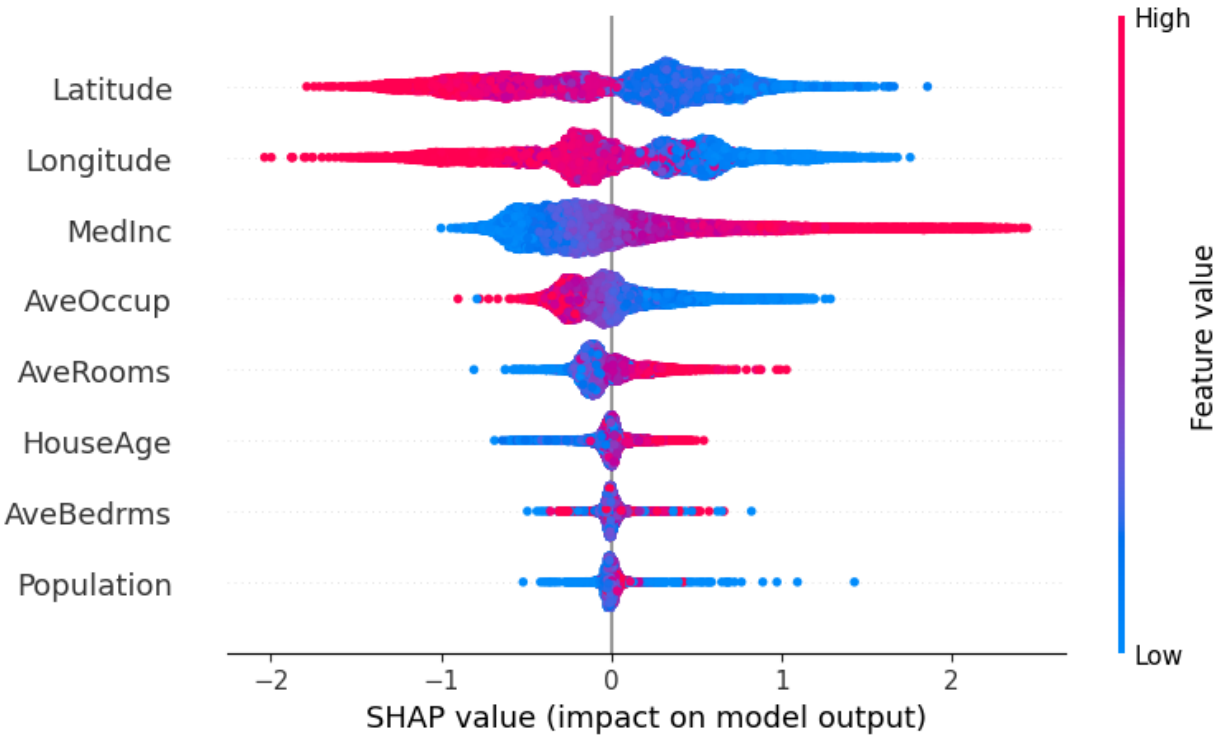

Рисунок 18 - Глобальная интерпретация модели машинного обучения, построенной на основе набора данных *california housing*, методом SHAP

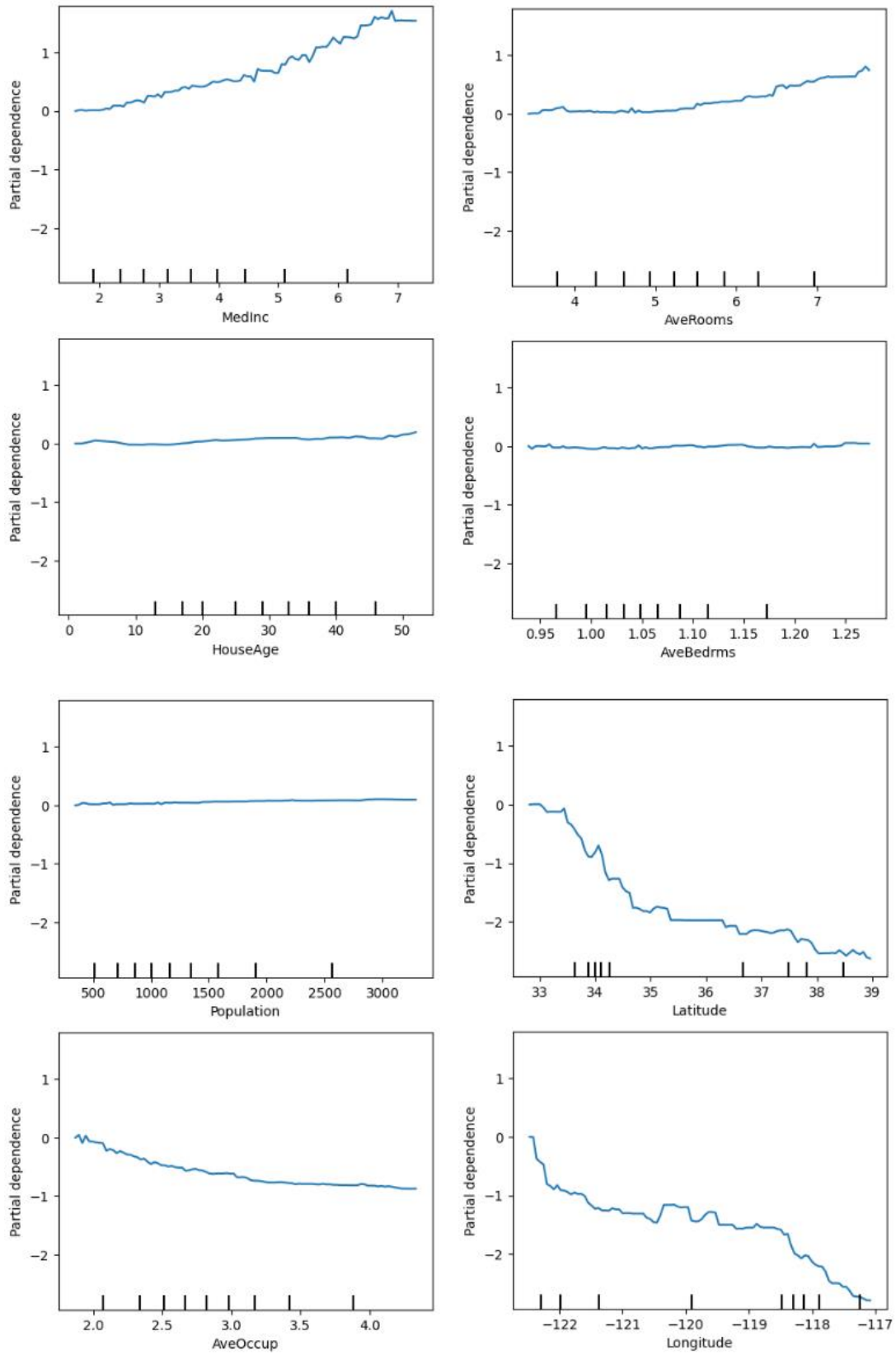

Рисунок 19 - Глобальная интерпретация модели машинного обучения, построенной на основе набора данных *california housing*, методом PDP

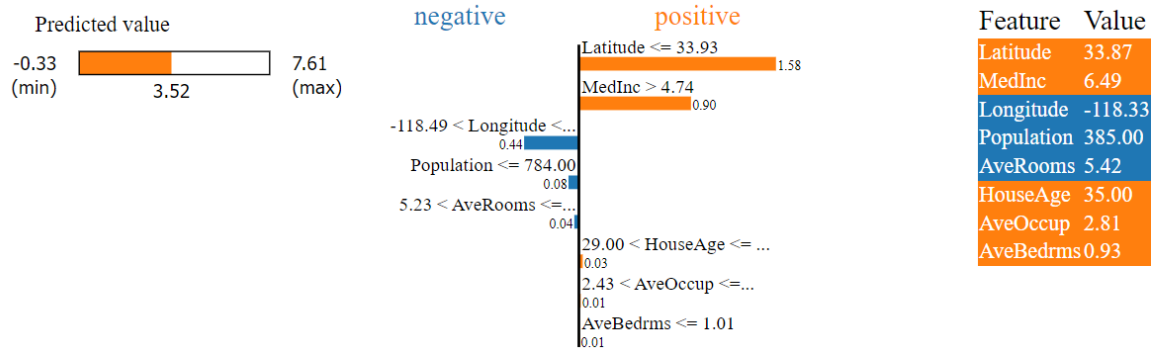

Рисунок 20 - Локальная интерпретация модели машинного обучения для объекта №50, построенной на основе набора данных *california housing*, методом LIME

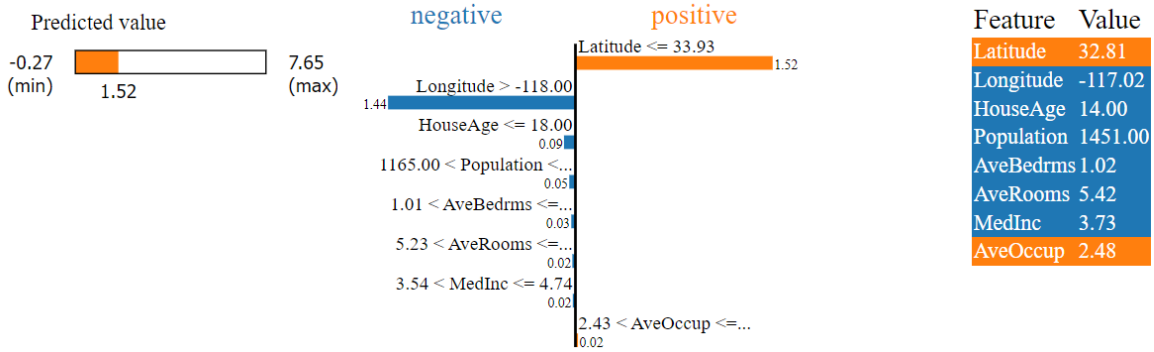

Рисунок 21 - Локальная интерпретация модели машинного обучения для объекта №100, построенной на основе набора данных *california housing*, методом LIME

3.3. Анализ результатов

3.3.1. Эксперимент 1

В результате эксперимента 1 были получены графики результатов интерпретации методами SHAP, PDP и LIME.

Из рисунка 12 можно сделать вывод, что самыми значимыми признаками, влияющими на развитие сахарного диабета в ближайшие пять лет, являются плазменные концентрации глюкозы в крови, индекс массы тела и возраст. Наименее значимыми признаками являются количество инсулина в крови и толщина кожи в области трицепса. Признаки оценка предрасположенности к диабету, диастолическое артериальное давление, количество инсулина в крови

вносят умеренный вклад в риск развития сахарного диабета в ближайшие пять лет у объекта.

Из рисунка 13 можно сделать вывод, что чем выше значение таких признаков как плазменные концентрации глюкозы в крови, индекс массы тела, возраст, оценка предрасположенности к диабету, количество беременностей и толщина кожи в области трицепса, тем выше риск развития сахарного диабета в течении пяти лет. При этом, чем выше значения признаков количество инсулина в крови и диастолическое артериальное давление, тем риск развития сахарного диабета в течении пяти лет ниже.

Из рисунка 14 можно сделать выводы аналогичные выводам, сделанным при анализе рисунка 12.

Из рисунка 15 можно сделать вывод, что несмотря на то что значения признаков оценка предрасположенности к диабету, диастолическое артериальное давление и количество инсулина в крови не приводят к риску развития сахарного диабета у объекта, значения остальных признаков показывают, что у объекта есть риск развития сахарного диабета в течении пяти лет.

Из рисунка 16 можно сделать вывод, что несмотря на то что значения признаков индекс массы тела и толщина кожи в области трицепса приводят к риску развития сахарного диабета у объекта, значения остальных признаков показывают, что у объекта нет риска развития сахарного диабета в течении пяти лет.

3.3.2. Эксперимент 2

В результате эксперимента 2 были получены графики результатов интерпретации методами SHAP, PDP и LIME.

Из рисунка 17 можно сделать вывод, что самыми значимыми признаками, влияющими медианную стоимость дома в квартале, являются широта квартала с недвижимостью, долгота квартала с недвижимостью, медианный доход в квартале и количество семей в квартале. Наименее значимыми признаками являются общее количество спален в квартале и население квартала. Признаки

общее количество комнат в квартале и медиана возраста домов в квартале вносят умеренный вклад в медианную стоимость дома в квартале.

Из рисунка 18 можно сделать вывод, что чем выше значение таких признаков как общее количество комнат в квартале, медиана возраста домов в квартале, медианный доход в квартале тем выше медианная стоимость дома в квартале. При этом, чем выше значения признаков широта квартала с недвижимостью, долгота квартала с недвижимостью, количество семей в квартале, тем ниже медианная стоимость дома в квартале. Значения признаков количество спален в квартале и население квартала почти не влияют на медианная стоимость дома в квартале.

Из рисунка 19 можно сделать выводы аналогичные выводам, сделанным при анализе рисунка 17.

Из рисунка 20 можно сделать вывод, что на повышение медианной стоимости дома в квартале повлияли все признаки кроме долготы квартала с недвижимостью, общего количество комнат в квартале и населения квартала. Они снизили медианную стоимость дома в квартале.

Из рисунка 21 можно сделать вывод, что на повышение медианной стоимости дома в квартале повлияли такие признаки как широта квартала с недвижимостью, население квартала, общее количество спален в квартале и количество семей в квартале. На понижение медианной стоимости дома в квартале повлияли остальные признаки.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы удалось произвести обзор основных методом машинного обучения и методы их интерпретации, а также их преимущества и недостатки.

В ходе выполнения работы были выбраны инструменты для реализации ПО на языке программирования *Python*, автоматизирующего процесс интерпретации моделей машинного обучения, а также реализовано ПО, автоматизирующее процесс интерпретации моделей машинного обучения.

Кроме того, были проведены вычислительные эксперименты, показавшие работоспособность реализованного ПО.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бринк Х., Машинное обучение / Х., Бринк, Д., Ричардс, М., Феверолф. – Санкт-Петербург : Питер, 2017. – 338 с.

2. Molnar Christoph. (2022). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable (2nd ed.). [Электронный ресурс] [Офиц. сайт]. URL: christophm.github.io/interpretable-ml-book/.

3. Артюшин Г. О., Коржаков Д. А., Хайрулин Т. Р. Обзор методов машинного обучения //Матрица научного познания. – 2020. – №. 6. – С. 27-31.

4. Кугаевских, А.В. Классические методы машинного обучения: Учебное пособие пособие / А.В. Кугаевских, Д.И. Муромцев, О.В. Кирсанова. – Санкт-Петербург : Редакционно-издательский отдел Университета ИТМО, 2022. – 53 с.

5. Логистическая регрессия (Logistic Regression) · Loginom Wiki. [Электронный ресурс] [Офиц. сайт]. URL: https://wiki.loginom.ru/articles/logistic-regression.html.

6. Лифшиц Ю. Метод опорных векторов. [Электронный ресурс] [Офиц. сайт]. URL: http://logic.pdmi.ras.ru/~ yura/internet/07ia.pdf.

7. Сизов А. А., Николенко С. И. Наивный байесовский классификатор // DOCPLAYER. [Электронный ресурс] [Офиц. сайт]. URL: https://docplayer.ru/45424867-Naivnyy-bayesovskiy-klassifikator.html.

8. Линейная регрессия: примеры и вычисление функции потерь. [Электронный ресурс] [Офиц. сайт]. URL: https://neurohive.io/ru/osnovy-data-science/linejnaja-regressija/.

9. Чистяков С. П. Случайные леса: обзор //Труды Карельского научного центра Российской академии наук. – 2013. – №. 1. – С. 117-136.

10. Гудфеллоу Я., Иошуа Б., Курвилль А. Глубокое обучение. – Litres, 2022.

11. Чару, А. Нейронные сети и глубокое обучение: учебный курс. / А. Чару,. – Санкт-Петербург : Диалектика, 2020. – 752 с.

12. Пичугин О. Н., Прокофьева Ю. З., Александров Д. М. Деревья решений как эффективный метод анализа и прогнозирования //Нефтепромысловое дело. – 2013. – №. 11. – С. 69-75.

13. Рассел, С., Искусственный интеллект. Современный подход / С., Рассел,, П., Норвиг,. – Москва : Вильямс, 2021. – 704 с.

14. lime Documentation, Release 0.1 [Электронный ресурс] [Офиц. сайт]. URL: https://lime-ml.readthedocs.io/en/latest/.

15. Documentation scikit-learn: machine learning in Python — scikit-learn 0.21.3 documentation [Электронный ресурс] [Офиц. сайт]. URL: https:// scikit-learn.org/0.21/documentation.html.

16. SHAP documentation [Электронный ресурс] [Офиц. сайт]. URL: https://shap.readthedocs.io/en/latest/index.html.

17. Explainable AI (XAI) Methods Part 3 — Accumulated Local Effects (ALE) [Электронный ресурс] [Офиц. сайт]. https://towardsdatascience.com/explainable-ai-xai-methods-part-3-accumulated-local-effects-ale-cf6ba3387fde.

18. InterpretML documentation [Электронный ресурс] [Офиц. сайт]. URL: https://interpret.ml/

19. pdpbox Documentation Release 0.2.0 [Электронный ресурс] [Офиц. сайт]. URL: https://readthedocs.org/projects/pdpbox/downloads/pdf/latest/

20. DALEX documentation [Электронный ресурс] [Офиц. сайт]. URL: https://dalex.drwhy.ai/python/api/

21. FairML documentation [Электронный ресурс] [Офиц. сайт]. URL: https://fairlearn.org/v0.8/contributor_guide/index.html

22. XGBoost Documentation — xgboost 1.5.0-SNAPSHOT documentation [Электронный ресурс] [Офиц. сайт]. URL: https://xgboost.readthedocs. io/en/latest/.

23. LightGBM Documentation v.3.3.2 [Электронный ресурс] [Офиц. сайт]. URL: https://lightgbm.readthedocs.io/en/v3.3.2/.

24. CatBoost Documentation [Электронный ресурс] [Офиц. сайт]. URL: https://catboost.ai/en/docs/.

25. scikit-learn documentation [Электронный ресурс] [Офиц. сайт]. URL: https://scikit-learn.org/0.21/documentation.html

26. PyCaret version 1.0.0 [Электронный ресурс] [Офиц. сайт]. URL: https://pycaret.readthedocs.io/en/stable/

27. H2O.ai documentation [Электронный ресурс] [Офиц. сайт]. URL: https://docs.h2o.ai/

28. PyALE documentation [Электронный ресурс] [Офиц. сайт]. URL: https://github.com/DanaJomar/PyALE

ПРИЛОЖЕНИЕ А. Текст программы

Листинг кода А.1 – класс *Inter*, автоматизирующий процесс интерпретации аппроксимирующих моделей машинного обучения.

```
class Inter():
    def __init__(self, data, metka, idd, model_type):
      self.data=data
      self.metka=metka
      self.idd=idd
      self.model_type=model_type

    def mod(self):
      global model
      global X_train
      global X
      global X_test
      global f
      X = self.data.drop([self.metka], axis=1)
      y = self.data[self.metka]
      X_train,     X_test,     y_train,     y_test     =
train_test_split(X, y, test_size = 0.3, random_state = 0)
      if self.model_type == 'R':
        model = XGBRegressor()
      else:
        model = XGBClassifier()
      model.fit(X_train, y_train)
      y_pred = model.predict(X)
      print('Модель обучена')

    def inter_global(self):
```

```
        explainer = shap.Explainer(model)
        shap_values = explainer(X_train)

        print('Синий цвет точки означает, что признак, на основе которого, расчитывается SHAP values имеет низкое значение, красный цвет - признак имеет высокое значение,')
        print('чем больше значение точки отображающей SHAP values, тем более значимым является признак')
        shap.summary_plot(shap_values, X_train, plot_type="dot")
        print('График отображает значимость признаков модели')
        shap.summary_plot(shap_values, X_train, plot_type="bar")

        print('График отображает значимость признаков модели')
        ftrs=self.data.keys()
        ftrs=ftrs.drop([self.metka])
        print(ftrs)
        features = ftrs
        fig, axs = plt.subplots(len(features), 1, figsize=(5, 35)) # индексы признаков, для которых строятся PDP both-PDP individual-ICE
        disp = PartialDependenceDisplay.from_estimator(model, X, features=features, kind='average',centered=True, ax=axs)


    def inter_local(self):
```

```
      if self.idd == None:
        print('Введите номер объекта')
      else:
        ftrs=self.data.keys()
        ftrs=ftrs.drop([self.metka])
        print('График отображает значимость признаков для
объекта №', self.idd)
        explainer                                          =
lime.lime_tabular.LimeTabularExplainer(X_train.values,
feature_names=X_train.columns.values.tolist(),

class_names=[self.metka],                    verbose=True,
mode='regression')
        exp                                                =
explainer.explain_instance(X_test.values[self.idd],
model.predict, num_features=len(ftrs))
        exp.show_in_notebook(show_table=True)
```

ПРИЛОЖЕНИЕ Б. Графическая часть ВКР

В графическую часть выпускной квалификационной работы входят:

- Слайд 1. Введение
- Слайд 2. Цели и задачи
- Слайд3. Модели машинного обучения
- Слайд 4. Методы интерпретации моделей машинного обучения
- Слайд 5. Метод LIME
- Слайд 6. Метод PDP
- Слайд 7. Метод SHAP
- Слайд 8. Блок-схема разработанного класса Inter
- Слайд 9. Вычислительные эксперименты
- Слайд 10. Эксперимент 1
- Слайд 11. Эксперимент 1
- Слайд 12. Эксперимент 1
- Слайд 13. Эксперимент 2
- Слайд 14. Эксперимент 2
- Слайд 15. Эксперимент 2
- Слайд 16. Заключение