**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*

«Тема»

| | | |
|---|---|---|
| Студент РК6-81Б (Группа) | | И.О. Фамилия |
| | (подпись, дата) | (инициалы и фамилия) |
| Руководитель ВКР | | А.П. Соколов |
| | (подпись, дата) | (инициалы и фамилия) |
| Консультант | | И.О. Фамилия |
| | (подпись, дата) | (инициалы и фамилия) |
| Нормоконтролер | | С.В. Грошев |
| | (подпись, дата) | (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

*Фамилия Имя Отчество*
(фамилия, имя, отчество)

Тема выпускной квалификационной работы

*Тема*

Источник тематики (кафедра, предприятие, НИР): кафедра

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

***Часть 1.*** *Аналитический обзор литературы*

*Более подробная формулировка задания. Следует сформировать, исходя из исходной постановки задачи, предоставленной руководителем изначально. Формулировка включает краткое перечисление задач, которые требовалось решить, включая, например: анализ существующих методов решения, выбор технологий разработки, обоснование актуальности тематики и др. Например: «В рамках аналитического обзора литературы должны быть изучены вычислительные методы, применяемые для решения задач кластеризации больших массивов данных. Должна быть обоснована актуальность исследований»*

***Часть 2.*** *Математическая постановка задачи / разработка архитектуры программной реализации / программная реализация*

*Более подробная формулировка задания. В зависимости от поставленной задачи: а) общая тема части может отличаться от работы к работе (например, может быть просто «Математическая постановка задачи» или «Архитектура программной реализации»), что определяется целесообразностью для конкретной работы; б) содержание задания должно несколько детальнее раскрывать заголовок. Примеры: 1. «Должна быть создана математическая модель распространения вирусной инфекции и представлена в форме*

*системы дифференциальных уравнений», 2. «Должно быть разработано веб-приложение, которое обеспечит ввод входных данных для модуля автоматизированной диагностики».*

***Часть 3.*** *Проведение вычислительных экспериментов, отладка и тестирование*

*Более подробная формулировка задания. Должна быть представлена некоторая конкретизация: какие вычислительные эксперименты требовалось реализовать, какие тесты требовалось провести для проверки работоспособности разработанных программных решений. Формулировка задания должна включать некоторую конкретику, например: какими средствами требовалось пользоваться для проведения расчетов и/или вычислительных экспериментo. Например: «Вычислительные эксперименты должны быть проведены с использованием разработанного в рамках ВКР программного обеспечения.»*

***Оформление выпускной квалификационной работы:***

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

| *[здесь следует ввести количество иллюстраций, чертежей, плакатов]* |
|---|
| |
| |
| |
| |

Дата выдачи задания «13» февраля 2023 г.

**Студент** И.О. Фамилия

(Подпись, дата) (И.О.Фамилия)

**Руководитель выпускной квалификационной работы** А.П. Соколов

(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: *Фамилия Имя Отчество*
(Фамилия, имя, отчество)

Тема выпускной квалификационной работы: [Тема]

| № 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 | *18.02.2023 Планируемая дата* | *18.02.2023* | Руководитель ВКР | *А.П. Соколов* |
| 2. | 1 часть: *аналитический обзор литературы* | *18.02.2023 Планируемая дата* | *31.03.2023* | Руководитель ВКР | *А.П. Соколов* |
| 3. | Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач | *28.02.2023 Планируемая дата* | *28.02.2023* | Заведующий кафедрой | *А.П. Карпенко* |
| 4. | 2 часть: *математическая постановка задачи, разработка архитектуры программной реализации, программная реализация* | *31.03.2023 Планируемая дата* | *31.03.2023* | Руководитель ВКР | *А.П. Соколов* |
| 5. | 3 часть: *проведение вычислительных экспериментов, отладка и тестирование* | *30.04.2023 Планируемая дата* | *30.04.2023* | Руководитель ВКР | *А.П. Соколов* |
| 6. | 1-я редакция работы | *31.05.2023 Планируемая дата* | *31.05.2023* | Руководитель ВКР | *А.П. Соколов* |
| 7. | Подготовка доклада и презентации | *07.06.2023 Планируемая дата* | *07.06.2023* | | |
| 8. | Заключение руководителя | *08.06.2023 Планируемая дата* | *08.06.2023* | Руководитель ВКР | *А.П. Соколов* |
| 9. | Допуск работы к защите на ГЭК (нормоконтроль) | *09.06.2023 Планируемая дата* | *09.06.2023* | Нормоконтролер | *С.В. Грошев* |
| 10 | Внешняя рецензия | *07.06.2023 Планируемая дата* | *07.06.2023* | | |
| 11 | Защита работы на ГЭК | *20.06.2023 Планируемая дата* | *20.06.2023* | | |

*Студент _____ И.О. Фамилия*
(подпись, дата)

*Руководитель работы _____ А.П. Соколов*
(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 *Фамилия Имя Отчество* группы *РК6-82Б*

на защиту выпускной квалификационной работы *Тема*
Декан факультета _____ «____» _____2023 г.

**Справка об успеваемости**

Студент *Фамилия Имя Отчество* за время пребывания в МГТУ имени Н.Э. Баумана с *2020* г. по *2023* г. полностью выполнил учебный план со следующими оценками: отлично – *процент* %, хорошо – *процент* %, удовлетворительно – *процент* %.

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. Соколов «____» _____ 2023 г.
(подпись) (ФИО) (дата)

Студент _____ И.О. Фамилия «____» _____ 2023 г.
(подпись) (ФИО) (дата)

# РЕФЕРАТ

**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)**

# АКТ
# проверки выпускной квалификационной работы

Студент группы *РК6-82Б*

*Фамилия Имя Отчество*

*(Фамилия, имя, отчество)*

Тема выпускной квалификационной работы: *[Тема]*

Выпускная квалификационная работа проверена, размещена в ЭБС «Банк ВКР» в полном объеме и соответствует / не соответствует требованиям, изложенным в Положении о порядке

*ненужное зачеркнуть*

подготовки и защиты ВКР.

Объем заимствования составляет _____% текста, что с учетом корректного заимствования соответствует / не соответствует требованиям к ВКР

_____

*ненужное зачеркнуть* *бакалавра, специалиста, магистра*

**Нормоконтролёр** _____ С.В. Грошев

(подпись) (ФИО)

Согласен:

**Студент** _____ И.О. Фамилия

(подпись) (ФИО)

Дата:

**Оглавление**

## 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 и решать сложные задачи. Однако, часто возникает необходимость не только в предсказании, но и в понимании процессов, лежащих в основе принятия решений модели. Именно в этой ситуации используются 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 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.

Основными характеристиками методов интерпретации моделей машинного обучения являются:

1. Признак;
2. Целевая переменная;
3. Входные данные;
4. Выходные данные.
5. Объект
6. 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 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 анализа, исследования и обучения моделей машинного обучения.

В рамках проводимой работы необходимо:

1. Изучить существующие модели машинного обучения, описать их преимущества и недостатки.

2. Изучить существующие подходы к интерпретации моделей машинного обучения, описать их преимущества и недостатки.

## 1.2. Обзор методов машинного обучения

Методы 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 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 данных, чтобы научиться предсказывать целевую переменную на основе ее связи с другими признаками.

### 1.2.1. Классические методы машинного обучения

Классические методы машинного обучения - это методы обучения моделей на основе статистических алгоритмов, которые используются для анализа и обработки данных [4]. Основные этапы построения математической модели классических методов машинного обучения:

1. Загрузка данных и подготовка их для анализа. Может включать в себя следующие шаги: очистка данных, нормализация и преобразование признаков.

2. Разделение выборки на обучающую и тестовую. Обучающая выборка используется для обучения модели, а тестовая - для проверки качества ее работы.
3. Обучение модели на обучающей выборке.
4. Проверка качества работы модели на тестовой выборке. Это позволяет оценить ее точность и определить, насколько хорошо она справляется с задачей классификации.

Основные преимущества классических методов машинного обучения:

1. Алгоритмы просты в реализации и не требуют большого количества параметров.
2. Работают с категориальными и числовыми признаками.
3. Простота интерпретации методов.

Основные недостатки классических методов машинного обучения:

1. Модели чувствительны к выбросам.
2. Модели не устойчивы к мультиколлинеарности.
3. Необходимость правильной подготовки данных.

**Логистическая регрессия**

Логистическая регрессия - это метод машинного обучения, который 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 метода логистической регрессии:

1. Хорошая обобщающая способность: логистическая регрессия показывает хорошие результаты на различных наборах данных, включая данные с большим числом признаков и небольшим

количеством образцов. Это делает ее хорошим выбором для широкого круга задач машинного обучения.

2. Гибкость: логистическая регрессия может быть адаптирована для решения многоклассовых задач классификации с использованием различных подходов.

Недостатки метода логистической регрессии:

1. Линейность: 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.

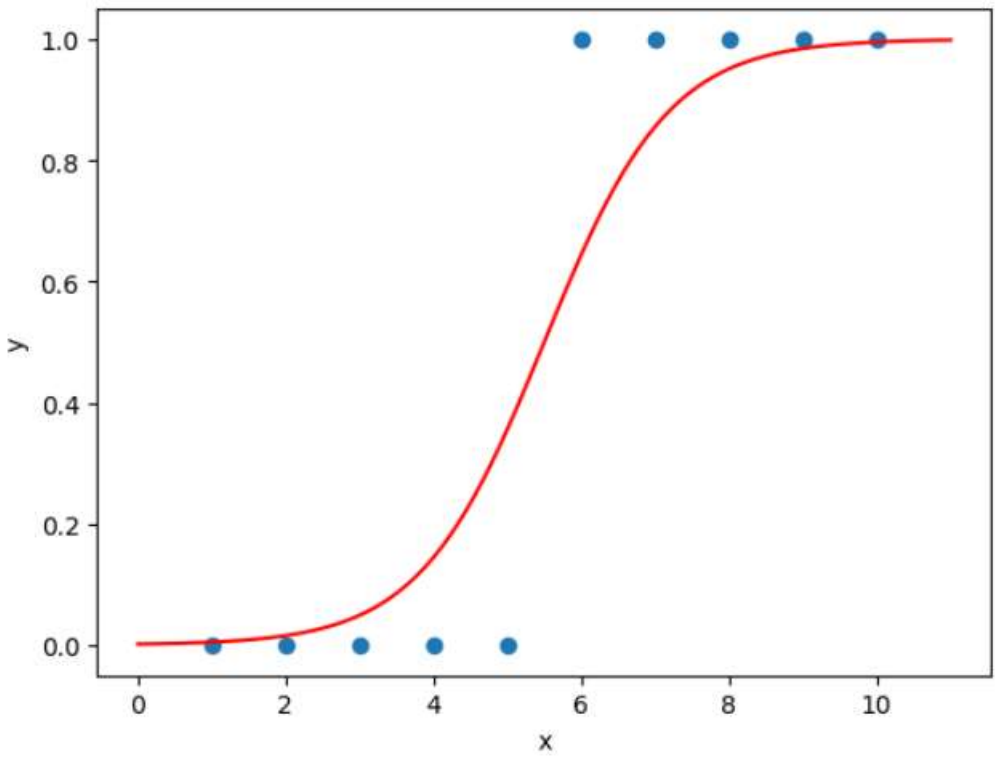


Рисунок 1. График логистической регрессии.

**Метод опорных векторов (Support Vector Machines, SVM)**

Метод опорных векторов - это алгоритм машинного обучения, используемый для классификации [6]. Он 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.

Преимущества метода опорных векторов являются:

1. Метод может работать с данными в любом $n$-мерном пространстве.
2. Метод эффективно обрабатывает данные с большим количеством признаков.

Недостатками метода опорных векторов могут быть:

1. Обучение может занять много времени для больших объемов данных.

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

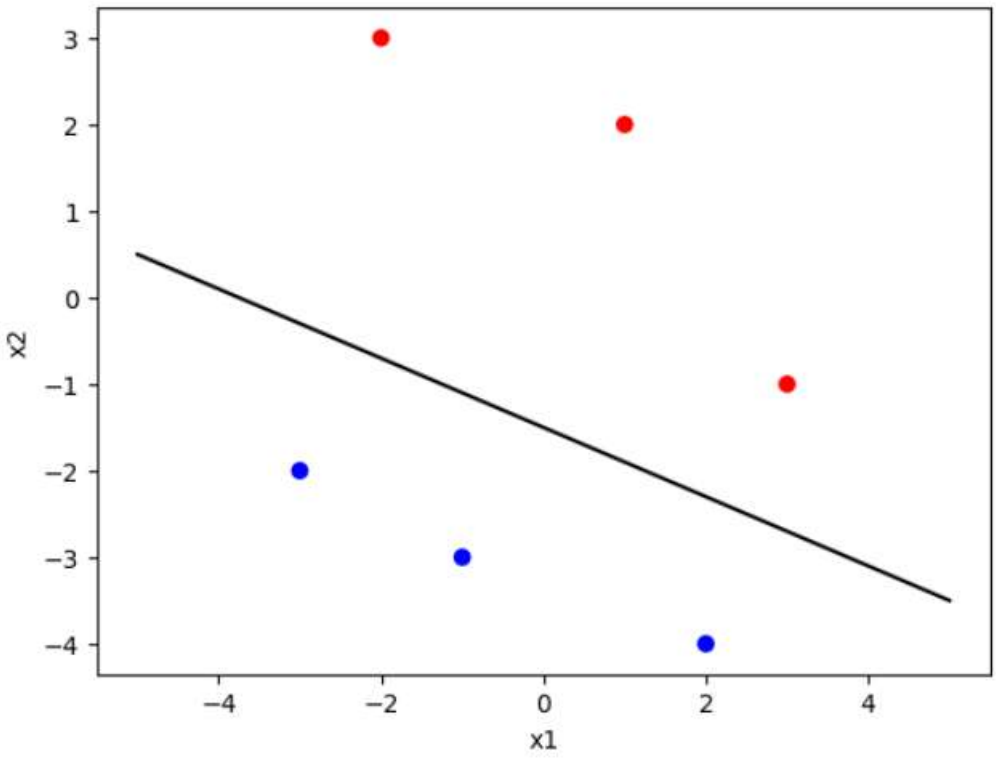


Рисунок 2. График SVM.

**Наивный байесовский классификатор**

Наивный байесовский классификатор — это метод машинного обучения, используемый для классификации данных [7]. Он основан 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. Важно отметить, что в данном методе предполагается, что признаки независимы друг от друга, что может быть неверным предположением в реальных данных.

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

Преимущества метода наивного байесовского классификатора:

1. Наивный байесовский классификатор требует меньше данных для обучения, чем многие другие алгоритмы машинного обучения, что делает его подходящим выбором для задач с ограниченным количеством данных.

Недостатки метода наивного байесовского классификатора:

1. Предположение о независимости признаков может быть неверным в реальных данных, что приводит к потере точности.
2. Если в выборке отсутствует объект с определенным значением признака, то вероятность принадлежности к данному классу будет равна нулю.

**Линейная регрессия**

Метод линейной регрессии – это метод машинного обучения, который используется для определения зависимости между независимыми переменными и зависимой переменной [8].

Суть метода заключается в нахождении линейной функции, которая наилучшим образом описывает зависимость между признаками и целевой переменной. Для этого используется метод наименьших квадратов, который минимизирует сумму квадратов отклонений предсказанных значений от реальных значений целевой переменной. Линейная регрессия может быть расширена до множественной линейной регрессии, где модель предсказывает целевую переменную на основе нескольких признаков.

Преимущества метода линейной регрессии:

1. Модель линейной регрессии может быть обучена быстро и эффективно.

Недостатки метода линейной регрессии:

1. Метод линейной регрессии требует выполнения предположений о распределении данных и связи между переменными, что может ограничивать его 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.

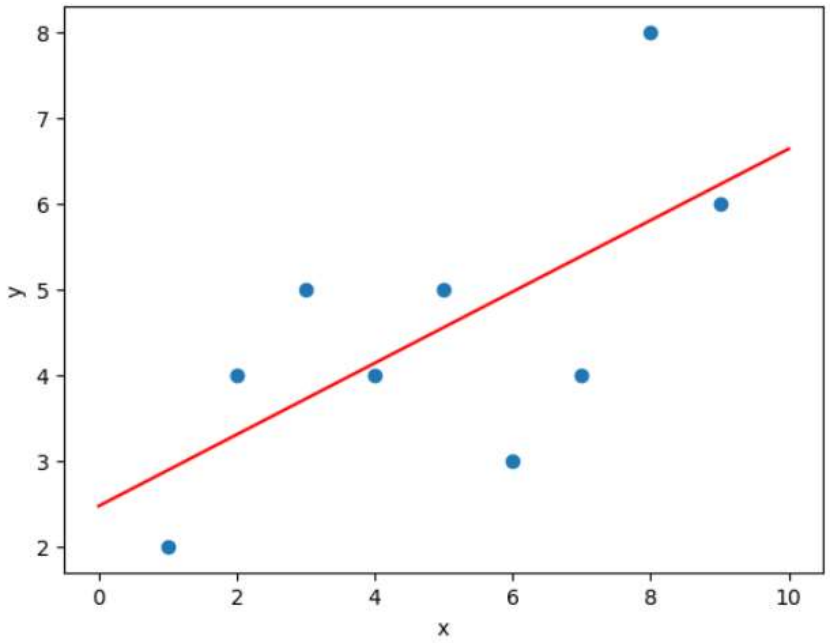


Рисунок 3. График линейной регрессии.

### 1.2.2. Глубокое обучение

Глубокое обучение - это раздел машинного обучения, который использует 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 сверточными, 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].

1. Подготовка данных: Данные являются ключевым элементом при обучении нейронной сети. Подготовка данных необходима для обучения и тестирования сети. Данный этап может включать в себя очистку, преобразование, масштабирование данных и разделение их на обучающий и тестовый наборы.
2. Создание модели нейронной сети: Модель нейронной сети определяет архитектуру сети и параметры ее компонентов. Эта модель строится на основе спецификации задачи и доступных данных.
3. Обучение нейронной сети: На данном этапе применяется выбранный алгоритм оптимизации, для обучения нейронной сети на обучающем наборе данных. Обучение происходит путем подачи входных данных в сеть, вычисления выходных значений, сравнения этих значений с ожидаемыми значениями и корректировки весов сети, чтобы минимизировать ошибку.
4. Тестирование и оценка производительности: После обучения проверяется производительность нейронной сети на тестовом наборе данных. Это позволяет оценить, насколько хорошо модель обобщает данные и может решать задачи, которые ранее не были ей известны.

Таким образом, нейронные сети - это метод машинного обучения, который требует нескольких шагов, начиная от подготовки данных и создания модели до обучения и тестирования, прежде чем он может быть использован для решения.

Преимущества нейронных сетей:

1. Способность обрабатывать сложные данные: Нейронные сети могут работать с данными большой размерности и могут выделять сложные зависимости между входными данными и выходными результатами.

2. Автоматическое извлечение признаков: В отличие от классических методов машинного обучения, нейронные сети могут автоматически извлекать признаки из данных, что упрощает процесс обучения.
3. Устойчивость к выбросам: Нейронные сети способны обрабатывать данные, которые содержат выбросы, благодаря своей способности обобщать и находить общие закономерности в данных.
4. Адаптивность к изменениям в данных: Нейронные сети могут обучаться на новых данных и адаптироваться к изменениям в данных без необходимости повторного обучения.

Недостатки нейронных сетей:

1. Высокая вычислительная сложность: Обучение нейронных сетей требует больших вычислительных мощностей и может занимать много времени.
2. Требуется много данных для обучения: Нейронные сети могут требовать большое количество данных для обучения, чтобы достичь хорошей производительности.
3. Неясность внутренней структуры: В отличие от классических методов машинного обучения, нейронные сети могут быть сложными для понимания и интерпретации из-за своей сложной структуры.
4. Подверженность переобучению: Если нейронная сеть обучается на слишком малом количестве данных или используется сложная модель, она может переобучиться и не обобщаться на новые данные.

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

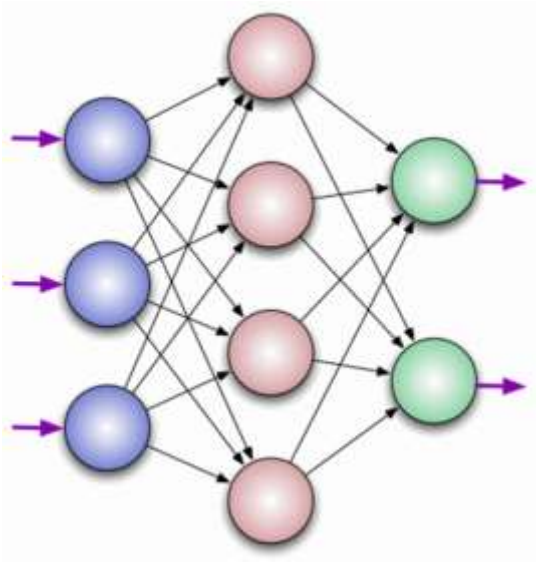

Рисунок 4. Пример нейронной сети.

### 1.2.3. Модели на основе решающих деревьев

**Решающие деревья**

Алгоритм решающих деревьев - это метод машинного обучения, используемый 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. Для выбора признака обычно используют меру неопределенности, которая показывает, насколько хорошо признак разделяет данные на классы или значения регрессии.
2. Разделение данных на две подгруппы на основе значения выбранного признака. Если признак является категориальным, то данные разделяются на подгруппы, соответствующие каждой категории признака. Если признак является числовым, то данные разделяются на подгруппы, в которых значение признака меньше или больше определенного порогового значения.

3. Повторение шагов 1-2 для каждой подгруппы, пока не будет достигнут критерий остановки. Критерием остановки может быть, например, достижение определенной глубины дерева, недостаточное увеличение неопределенности при разделении данных или недостаточное количество элементов в листовых узлах.
4. Присвоение классов или значений регрессии листовым узлам. Для классификации классом листового узла может быть класс, который наиболее часто встречается в подгруппе данных, которая достигает этого листового узла. Для регрессии значение листового узла может быть средним или медианным значением целевой переменной в подгруппе данных, которая достигает этого листового узла.

Преимущества метода решающих деревьев:

1. Решающие деревья легко интерпретировать, что позволяет анализировать причинно-следственные связи и принимать обоснованные решения.
2. Обучение решающего дерева происходит быстро и легко масштабируется для больших наборов данных.
3. Решающие деревья хорошо работают с большим количеством признаков, что позволяет использовать их в широком спектре задач.
4. Решающие деревья могут работать с данными разного типа, включая категориальные и числовые.

Недостатки метода решающих деревьев:

1. Решающие деревья могут быть склонны к переобучению, особенно если они слишком глубокие или если данные содержат выбросы.
2. Решающие деревья могут быть неустойчивы к небольшим изменениям в данных, что может привести к большим изменениям в структуре дерева.
3. Решающие деревья могут не справиться с задачами, где есть сложные зависимости между признаками.

4. Решающие деревья 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.

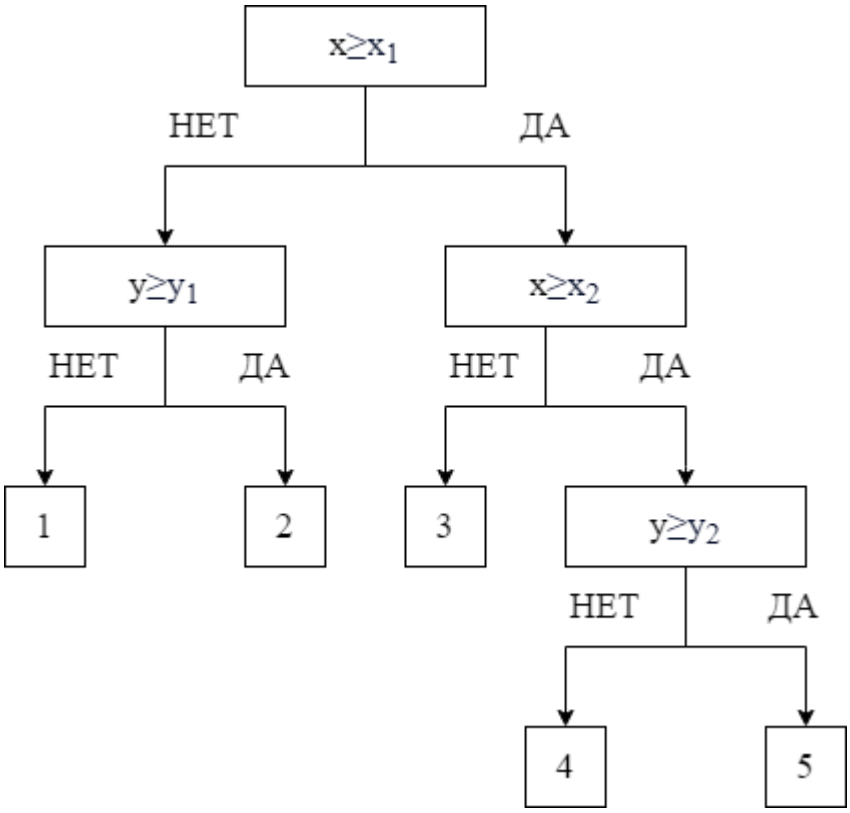


Рисунок 5. Пример решающего дерева.

Метод 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**

Метод 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 метода 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 метода 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 метода случайного леса:

1. Метод обладает высокой точностью и способностью к обобщению на новые данные.
2. Он может обрабатывать большие объемы данных и признаков.

Недостатки метода случайного леса:

1. Обучение ансамбля деревьев может занимать много времени.

2. Когда число признаков очень велико, метод может потерять в точности, так как случайное подмножество признаков может не содержать наиболее информативные признаки.
3. Метод сложно интерпретируемый, так как он использует множество деревьев решений.

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

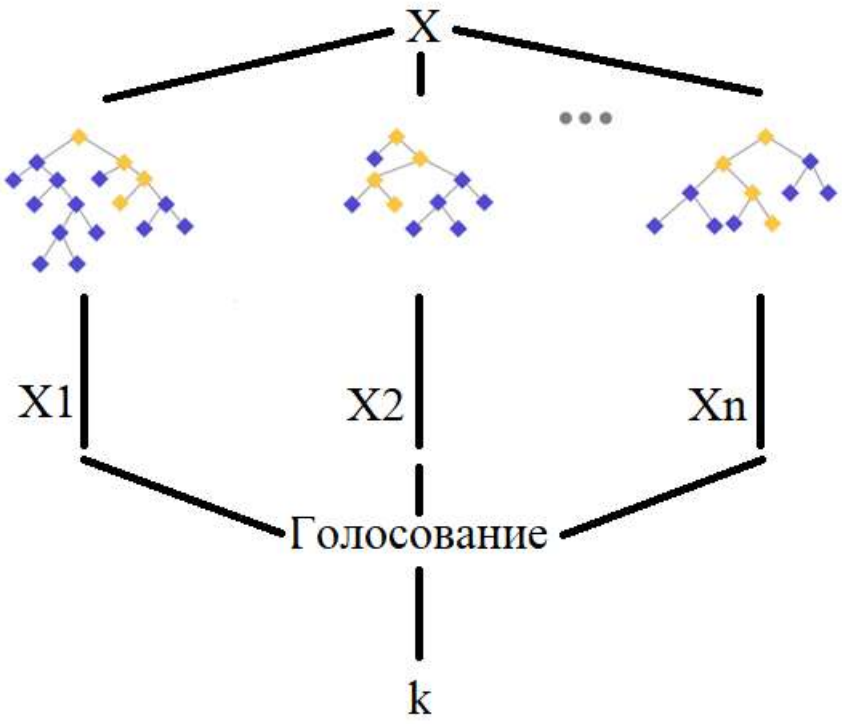


Рисунок 6. Пример случайного леса.

**Бустинг**

Метод бустинга решающих деревьев (boosting decision trees) - 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 модели и более успешно классифицировать те элементы данных, которые были неправильно классифицированы.
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 метода бустинга решающих деревьев:

1. Метод бустинга решающих деревьев обладает высокой точностью классификации или регрессии, что делает его эффективным методом машинного обучения.
2. Бустинг решающих деревьев может обрабатывать большие объемы данных и обучаться на данных с большим количеством признаков.
3. Метод бустинга решающих деревьев хорошо обрабатывает выбросы и шум в данных.
4. Бустинг решающих деревьев не требует предварительной обработки данных.

Недостатки метода бустинга решающих деревьев:

1. При использовании сложных моделей бустинга с большим числом деревьев, алгоритм может стать склонным к переобучению на тренировочных данных.
2. Выбросы в данных могут значительно повлиять на построение каждого нового дерева, так как алгоритм стремится минимизировать ошибку на всех данных.

3. Бустинг является вычислительно сложным алгоритмом, особенно когда используется большое количество деревьев или признаков.

## 1.3. Обзор методов интерпретации модели

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

Системы искусственного интеллекта используются во многих сферах обеспечения жизни человека [13]. С каждым годом интеллектуальные системы обрабатывают все больше данных и принимают все больше решений. Эти решения оказывают значимое влияние на людей.

Проблемой становится недоверие к полностью нечеловеческим, автономным системам искусственного интеллекта. Недоверие заключается в непонимании того, почему интеллектуальные системы принимают то или иное решение, исходя из каких убеждений такие системы действуют. Для решения проблемы «недоверия» к таким системам стали применять методы объяснимого искусственного интеллекта, называемые методами интерпретации моделей.

Для решения задачи интерпретации методов 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. Необходимо выбрать модель машинного обучения, которую нужно проинтерпретировать, и обучить ее на тренировочных данных.

2. Сбор выборки данных для интерпретации. Выборка должна содержать объекты данных, на которых модель должна делать предсказания, а также соответствующие им значения признаков.
3. Применение метода интерпретации. Необходимо применить выбранный метод интерпретации для получения интерпретируемых результатов. В результате этого шага получается набор правил, объясняющих, как модель принимает решения на основе входных данных.
4. Визуализация результатов. Полученные результаты необходимо визуализировать для более наглядного представления правил и их влияния на результаты модели.

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 наблюдения или экземпляра данных. Такие методы могут помочь в определении, почему модель приняла конкретное решение для данного наблюдения, и какие изменения в признаках могут привести к изменению решения.

**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 будет объяснять прогноз модели на конкретном объекте. Для этого метод LIME случайным образом создает выборку 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, используя линейную регрессию или другой алгоритм. 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. Следует отметить, что метод LIME не всегда гарантирует правильность интерпретации, особенно если выбраны неправильные признаки для образца.

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.

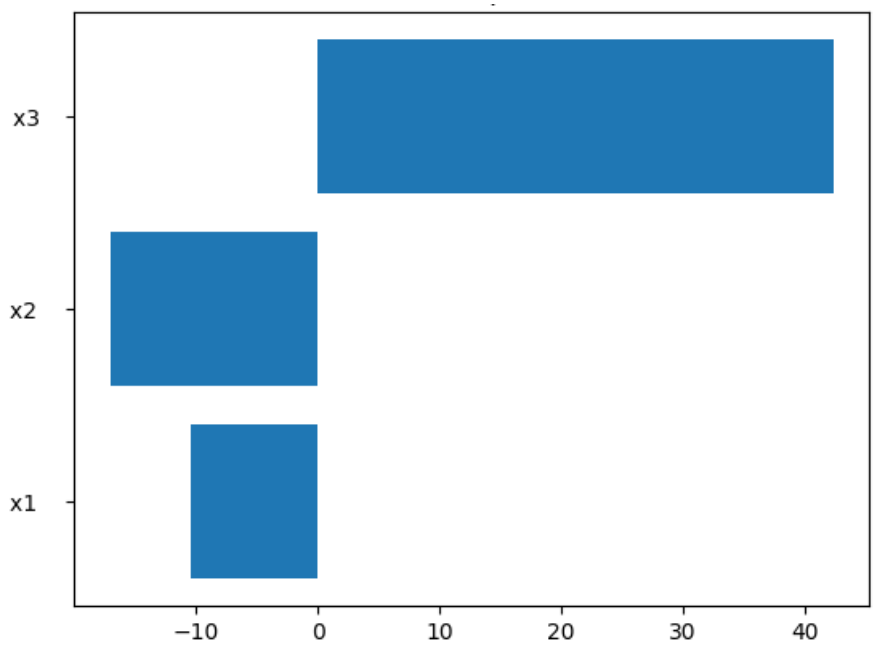


Рисунок 7. Пример интерпретации методом LIME.

**ICE (Individual Conditional Expectation)**

Метод ICE (Individual Conditional Expectation) - 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 и индивидуальные 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.

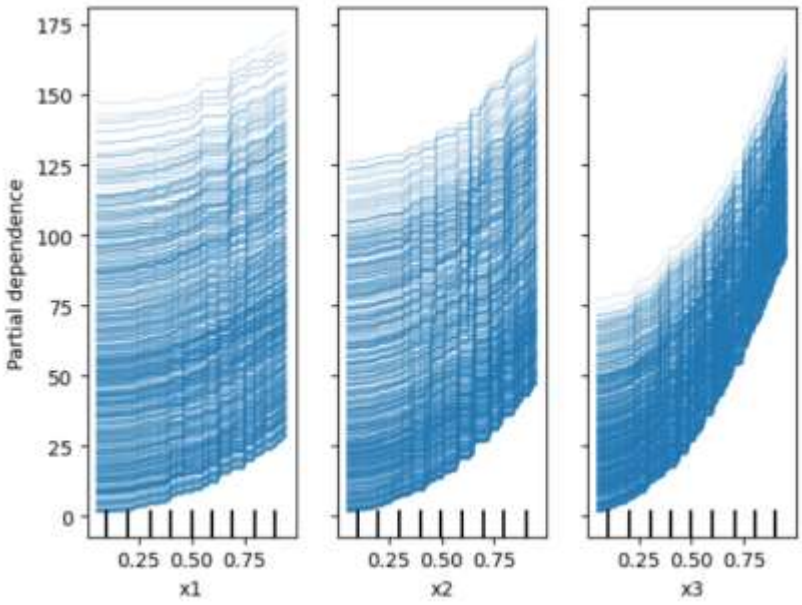


Рисунок 8. Пример интерпретации методом ICE.

**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).

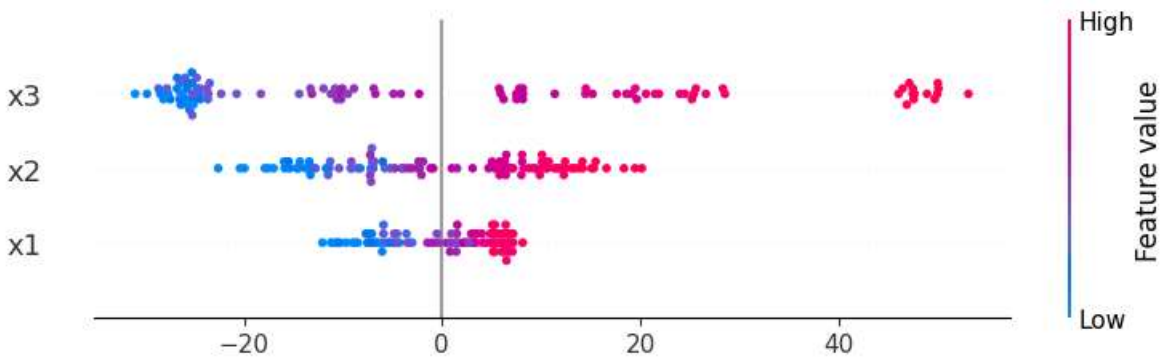


Рисунок 9. Пример интерпретации методом SHAP.

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, а также анализировать 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 привлекательным 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, 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. Для каждого признака строится график, на котором по оси X отображаются значения данного признака, а по оси Y - средние значения прогноза модели для всех объектов при соответствующих значениях всех остальных признаков.

Основное преимущество метода PDP заключается в его способности оценивать вклад каждого признака на прогноз модели, учитывая влияние всех остальных признаков, что позволяет получать более точные и универсальные интерпретации моделей машинного обучения. Кроме того, метод PDP 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, необходимость предварительной обработки данных и оптимизации гиперпараметров модели для достижения наилучших результатов интерпретации.

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).

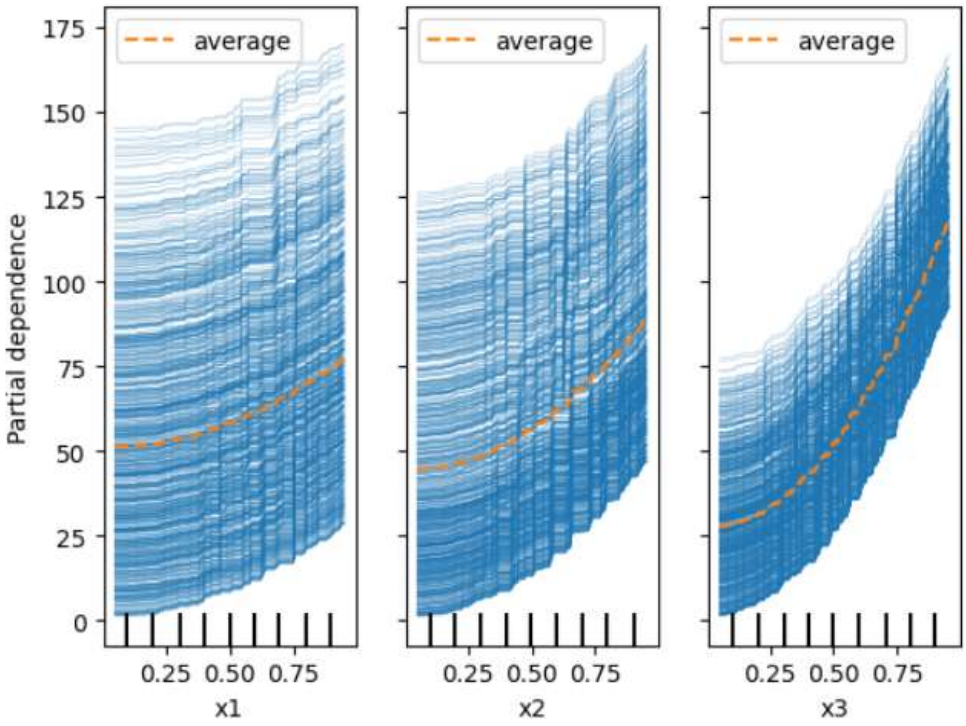


Рисунок 10. Пример интерпретации методом PDP.

**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 (аккумулируются) 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**

1. 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.
2. 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**

1. pdpbox - это библиотека для машинного обучения на Python, которая предоставляет инструменты для вычисления и визуализации ICE [19].
2. 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.
3. FairML - это библиотека на Python для интерпретации моделей машинного обучения, которая предоставляет инструменты для вычисления и визуализации ICE [21].

**SHAP**

Существует несколько библиотек ПО, которые реализуют метод SHAP, включая:

1. SHAP (Python): 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.
2. XGBoost SHAP (Python): 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.
3. LightGBM SHAP (Python): 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.
4. CatBoost SHAP (Python): 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**

1. Scikit-learn - это одна из самых популярных библиотек для машинного обучения на Python. Она также предоставляет функционал для вычисления и визуализации PDP [25].
2. 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].

3. pdpbox - это библиотека для машинного обучения на Python, которая предоставляет инструменты для вычисления и визуализации PDP. Она поддерживает модели, созданные с помощью библиотек Scikit-learn, XGBoost, CatBoost и LightGBM.
4. 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.

**ALE**

Существует несколько библиотек и фреймворков, которые могут быть использованы для визуализации ALE:

1. PyALE - это библиотека на Python, которая позволяет визуализировать ALE. Она имеет простой интерфейс и может быть использована для любой модели машинного обучения [28].
2. pdpbox - это библиотека на Python, которая включает функционал для визуализации ALE. Она может быть использована для моделей, созданных с помощью различных библиотек для машинного обучения, таких как scikit-learn или xgboost.
3. DALEX - это открытый и бесплатный фреймворк для интерпретации моделей машинного обучения на Python и R. DALEX включает функционал для вычисления ALE, который может быть использован вместе с моделями, созданными с помощью различных библиотек для машинного обучения.
4. scikit-learn - это библиотека на Python для машинного обучения, которая включает функционал для вычисления и визуализации ALE.

У разных методов интерпретации моделей есть разные слабые места, то есть эти методы могут давать ошибочные результаты на разных данных.

Результатом работы этих методов являются коэффициенты значимости каждого из признаков. Для повышения надежности коэффициентов необходимо разработать ПО, которое позволит агрегировать результаты работы разных методов интерпретации и повысить надежность коэффициентов

# 2. Методика интерпретации моделей машинного обучения на основе решающих деревьев

## 2.1. Математическое описание метода решающих деревьев

### 2.1.1. Определение решающего дерева

Рассмотрим бинарное дерево, в котором:

- каждой внутренней вершине $v$ приписан предикат $B_v: X \to 0,1$;
- каждой листовой вершине $v$ приписан прогноз $c_v \in Y$, где $Y$ — область значений целевой переменной (в случае классификации листу может быть также приписан вектор вероятностей классов).

В ходе предсказания осуществляется проход по этому дереву к некоторому листу. Для каждого объекта выборки $x$ движение начинается из корня. В очередной внутренней вершине $v$ проход продолжится вправо, если $B_v(x) = 1$, и влево, если $B_v(x) = 0$. Проход продолжается до момента, пока не будет достигнут некоторый лист, и ответом алгоритма на объекте $x$ считается прогноз $c_v$, приписанный этому листу.

Предикат $B_v$ может иметь, произвольную структуру, но, как правило, на практике используют сравнение с порогом $t \in R$ по произвольному $j$-му признаку:

$$B_v(x, j, t) = [x_j \leq t]. \quad (1)$$

При проходе через узел дерева с данным предикатом объекты будут отправлены в правое поддерево, если значение $j$-го признака у них меньше либо равно $t$, и в левое — если больше. В дальнейшем по умолчанию используются именно такие предикаты.

### 2.1.2. Построение дерева

Опишем базовый жадный алгоритм построения бинарного решающего дерева.

Начиная со всей обучающей выборки $X$, находится наилучшее, с точки зрения заранее заданного функционала качества $Q(X, j, t)$, её разбиение на две части $R_1(j,t) = \{x \mid x_j < t\}$ и $R_2(j,t) = \{x \mid x_j > t\}$. Найдя наилучшие значения $j$ и $t$, создается корневая вершина дерева, затем ей в соответствие ставится предикат $[x_j < t]$. Объекты разбиваются на две части — одни попадают в левое поддерево, другие в правое. Для каждой из этих подвыборок рекурсивно повторяется процедуру, строя дочерние вершины для корневой и последующих вершин. В каждой вершине проверяется, не выполнилось ли некоторое условие остановки — и если выполнилось, то прекращается рекурсия и эта вершина объявляется листом. Когда дерево построено, каждому листу ставится в соответствие ответ. В случае с классификацией это может быть класс, к которому относится больше всего объектов в листе, или вектор вероятностей. Для регрессии это может быть среднее значение, медиана или другая функция от целевых переменных объектов в листе. Выбор конкретной функции зависит от функционала качества в исходной задаче.

Решающие деревья могут обрабатывать пропущенные значения — ситуации, в которых для некоторых объектов неизвестны значения одного или нескольких признаков. Для этого необходимо модифицировать процедуру разбиения выборки в вершине.

После того, как дерево построено, можно провести его стрижку — удаление некоторых вершин с целью понижения сложности и повышения обобщающей способности..

Таким образом, метод построения решающего дерева определяется:

1. Видом предикатов в вершинах;
2. Функционалом качества Q(X, j, t);
3. Критерием остановки;

4. Методом обработки пропущенных значений;
5. Методом стрижки.

**Критерии информативности**

При построении дерева необходимо задать функционал качества, на основе которого осуществляется разбиение выборки на каждом шаге. Обозначим через $\boldsymbol{R_m}$ множество объектов, попавших в вершину, разбиваемую на данном шаге, а через $\boldsymbol{R_\ell}$ и $\boldsymbol{R_r}$ — объекты, попадающие в левое и правое поддерево соответственно при заданном предикате. Будем использовать функционалы следующего вида:

$$Q(R_m, j, s) = H(R_m) - \frac{|R_\ell|}{|R_m|} H(R_\ell) - \frac{|R_\ell|}{|R_m|} H(R_r). \quad (2)$$

Здесь $H(R)$ — это критерий информативности, который оценивает качество распределения целевой переменной среди объектов множества $R$. Чем меньше разнообразие целевой переменной, тем меньше должно быть значение критерия информативности — и, соответственно, будет минимизироваться его значение. Функционал качества $Q(R_m, j, s)$ при этом максимизируется.

Так как в каждом листе дерево будет выдавать константу — вещественное число, вероятность или класс, можно предложить оценивать качество множества объектов $R$ тем, насколько хорошо их целевые переменные предсказываются константой (при оптимальном выборе этой константы):

$$H(R) = \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} L(y_i, c), \quad (3)$$

где $L(y, c)$ — некоторая функция потерь.

**Регрессия**

В регрессии в качестве функции потерь выбирается квадрат отклонения:

$$L(y_i, c) = (y_i - c)^2. \quad (4)$$

В этом случае критерий информативности будет выглядеть как

$$H(R) = \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - c)^2 . \quad (5)$$

Минимум в этом выражении будет достигаться на среднем значении целевой переменной. Значит, критерий можно переписать в следующем виде:

$$H(R) = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} \left( y_i - \frac{1}{|R|} \sum_{(x_i, y_i) \in R} y_j \right)^2 . \quad (6)$$

Можно сделать вывод, что информативность вершины измеряется её дисперсией. Необходимо выбирать такие предикаты, чтобы сумма дисперсий в листьях была как можно меньше. Также можно использовать другие функции ошибки $L$, например, при выборе абсолютного отклонения в качестве критерия получается среднее абсолютное отклонение от медианы.

**Классификация**

Обозначим через $p_k$ долю объектов класса $k$ ($k \in \{1, \ldots, K\}$), попавших в вершину $R$:

$$p_k = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i = k] . \quad (7)$$

Через $k_*$ обозначим класс, чьих представителей оказалось больше всего среди объектов, попавших в данную вершину: $k_* = arg \max_k p_k$ .

**Ошибка классификации**

Рассмотрим индикатор ошибки как функцию потерь:

$$H(R) \min_{c \in Y} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i \neq c] . \quad (8)$$

Можно заметить, что оптимальным предсказанием будет наиболее частотный класс $k_*$ — значит, критерий будет равен следующей доле ошибок:

$$H(R) = \frac{1}{|R|} \sum_{(xi, yi) \in R} [y_i \neq k_*] = 1 - p_{k_*} .$$

Данный критерий является достаточно грубым, поскольку учитывает частоту $p_{k_*}$ лишь одного класса.

**Критерий Джини**

Рассмотрим ситуацию, в которой выдаём в вершине не один класс, а распределение на всех классах $c = (c_1, \ldots, c_K)$, $\sum_{k=1}^{K} c_k = 1$. Качество такого распределения можно измерять с помощью критерия Бриера:

$$H(R) = \min_{\sum_k c_k = 1} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} \sum_{k=1}^{K} (c_k - [y_i = k])^2 . \qquad (9)$$

Можно показать, что оптимальный вектор вероятностей состоит из долей классов $p_k$: $c_* = (p_1, \ldots, p_K)$.

Если подставить эти вероятности в исходный критерий информативности и провести ряд преобразований, то получается критерий Джини:

$$H(R) = \sum_{k=1}^{K} p_k (1 - p_k) . \qquad (10)$$

**Энтропийный критерий**

Способ оценивания качества вероятностей логарифмическими потерями:

$$H(R) \min_{\sum_k c_k = 1} \left( -\frac{1}{|R|} \sum_{(x_i, y_i) \in R} \sum_{k=1}^{K} [y_i = k] \log c_k \right). \qquad (11)$$

Для вывода оптимальных значений $c_k$ необходимо вспомнить, что все значения $c_k$ должны суммироваться в единицу. Из методов оптимизации известно, что для учёта этого ограничения необходимо искать минимум лагранжиана:

$$L(c, \lambda) = -\frac{1}{|R|} \sum_{(x_i, y_i) \in R} \sum_{k=1}^{K} [y_i = k\,] \log c_k + \lambda \sum_{k=1}^{K} c_k . \qquad (12)$$

Дифференцируя, получаем:

$$\frac{\partial}{\partial c_k} L(c, \lambda) = -\frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i = k] \frac{1}{c_k} + \lambda = -\frac{p_k}{c_k} + \lambda = 0,$$

откуда выражаем $c_k = \frac{p_k}{\lambda}$. Суммируя эти равенства по $k$, получим

$$\sum_{k=1}^{K} c_k = \frac{1}{\lambda}\sum_{k=1}^{K} p_k = \frac{1}{\lambda} = 1\,, \tag{13}$$

откуда $\lambda = 1$. Значит, как и в предыдущем случае, минимум достигается при $c_k = p_k$. Подставляя эти выражения в критерий, получим, что он будет представлять собой энтропию распределения классов:

$$H(R) = -\sum_{k=1}^{K} p_k \log p_k. \tag{14}$$

Из теории вероятностей известно, что энтропия ограничена снизу нулем, причем минимум достигается на вырожденных распределениях ($p_i = 1,\ p_j = 0$ для $i \neq j$). Максимальное же значение энтропия принимает для равномерного распределения. Можно сделать вывод, что энтропийный критерий отдает предпочтение более «вырожденным» распределениям классов в вершине.

**Критерии остановки**

Некоторые ограничения и критерии:

• 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.

С помощью грамотного выбора подобных критериев и их параметров можно существенно повлиять на качество дерева. Тем не менее, такой подбор является трудозатратным и требует проведения кроссвалидации.

**Обработка пропущенных значений**

Одним из основных преимуществ решающих деревьев является возможность работы с пропущенными значениями. Рассмотрим некоторые варианты.

Пусть нужно вычислить функционал качества для предиката

$\beta(x) = [x_j < t]$, в выборке $R$ для некоторых объектов неизвестно значение признака $j$ — обозначим эти объекты как $V_j$. В таком случае при вычислении функционала можно проигнорировать эти объекты, сделав поправку на потерю информации от этого:

$$Q(R, j, s) \approx \frac{|R \setminus V_j|}{|R|} Q(R\ V_j, j, s). \quad (15)$$

Затем, если данный предикат окажется лучшим, объекты из $V_j$ помещаются как в левое, так и в правое поддерево. Также можно присвоить этим объектам веса $|R_\ell|/|R|$ в левом поддереве и $|R_r|/|R|$ в правом. В дальнейшем веса можно учитывать, добавляя их как коэффициенты перед индикаторами $[y_i = k]$ во всех формулах.

На этапе применения дерева необходимо выполнять схожие действия. Если объект попал в вершину, предикат которой не может быть вычислен из-за пропуска, то прогнозы для него вычисляются в обоих поддеревьях, и затем усредняются с весами, пропорциональными числу обучающих объектов в этих поддеревьях. То есть, если прогноз вероятности для класса $k$ в поддереве $R_m$ обозначается через $a_{mk}(x)$, то получается:

$$a_{mk}(x) = \begin{cases} a_{\ell k}(x), & \beta_m(x) = 0; \\ a_{rk}(x), & \beta_m(x) = 1; \\ \frac{|R_\ell|}{|R_m|} a_{\ell k}(x) + \frac{|R_r|}{|R_m|} a_{rk}(x), & \beta_m(x) \text{ нельзя вычислить.} \end{cases} \quad (16)$$

Другой подход заключается в построении суррогатных предикатов в каждой вершине. Суррогатный предикат - предикат, который использует другой признак, но при этом дает разбиение, максимально близкое к данному. Зачастую схожее качество показывают и гораздо более простые способы обработки пропусков, например, можно заменить все пропуски на ноль. Для деревьев также можно заменить пропуски в признаке на числа, которые превосходят любое значение данного признака. В этом случае в дереве можно будет выбрать такое разбиение по этому признаку, что все объекты с

известными значениями пойдут в левое поддерево, а все объекты с пропусками — в правое.

**Методы стрижки дерева**

Одним из методов стрижки является механизм отсечения дерева (cost-complexity pruning). Дерево, полученное в результате работы жадного алгоритма, обозначается через $T_0$. Так как в каждом из листьев находятся объекты только одного класса, значение функционала $R(T)$ будет минимально на самом дереве $T_0$ (среди всех поддеревьев). Однако, данный функционал характеризует качество дерева на обучающей выборке, и чрезмерная подгонка под нее может привести к переобучению. Чтобы решить эту проблему, вводится новый функционал $R_a(T)$, представляющий собой сумму исходного функционала $R(T)$ и штрафа за размер дерева:

$$R_a(T) = R(T) + a|T|, \quad (17)$$

где $|T|$ — число листьев в поддереве $T$, а $a \geq 0$ — параметр. Это один из примеров регуляризованных критериев качества, которые ищут баланс между качеством классификации обучающей выборки и сложностью построенной модели.

Можно показать, что существует последовательность вложенных деревьев с одинаковыми корнями:

$$T_K \subset T_{K-1} \subset \cdots \subset T_0. \quad (18)$$

Здесь $T_K$ — тривиальное дерево, состоящее из корня дерева $T_0$, в котором каждое дерево $T_i$ минимизирует функционал (17) для $a$ из интервала $a \in [a_i, a_{i+1})$, причем $0 = a_0 < a_1 < \cdots < a_K < \infty$.

Эту последовательность можно эффективно найти путем обхода дерева. Далее из нее выбирается оптимальное дерево по отложенной выборке или с помощью кроссвалидации.

## 2.2. 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.

### 2.2.1. Shapley values в теории игр

Теория игр - это область математики, изучающей взаимодействие между игроками, преследующими некие цели и действующими по определенным правилам. Кооперативной игрой называется игра, в которой группа игроков (коалиция) действует совместно. С середины XX века известны так называемые 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, действующей совместно. 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 $S$:

$$\Delta(i, s) = (S \cup i) - v(S), \tag{19}$$

Пусть всего есть $N$ игроков. Рассмотрим множество П всех возможных упорядочиваний игроков, и обозначим за $p$ множество игроков, стоящих перед игроков $i$ в упорядочивании $\pi$. Shapley value для игрока $i$ рассчитывается таким образом:

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

То есть считается средний прирост эффективности от добавления $i$-го игрока в коалицию игроков, стоящих перед ним, по всем возможным упорядочиваниям игроков (количество элементов суммы равно $N!$).

Формула (20) задается аксиоматически, то есть формулируется ряд необходимых свойств и доказывает, что данное решение является единственным, которое удовлетворяет этим свойствам. 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:

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

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 $|S| \approx \frac{|N|}{2}$.

### 2.2.2. Shapley regression values

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

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:

$$v(S) = f_S(x_S). \tag{22}$$

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:

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

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 сранивается текущее значение признака на примере $x$ с его полным отсутствием при обучении и тестировании. Минусом

такого подхода является высокая сложность вычислений: для расчета Shapley regression values нужно обучать модель на всех возможных подмножествах признаков, что в большинстве случаев невыполнимо. Однако можно эффективнее посчитать приблизительное значение Shapley values, если считать не все элементы суммы по формуле (21), а лишь некоторые, обладающие большими весами. Для этого можно использовать веса как вероятности при семплировании элементов суммы [29].

### 2.2.3. SHAP values

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 SHAP values.** Пусть имеется модель $f$, распределение данных и некий тестовый пример $x$. Необходимо оценить важность текущих значений каждого признака по сравнению с их неопределенными значениями. 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* — это эффективная опенсорсная реализация алгоритма градиентного бустинга, специализирующаяся на построении

моделей на основе решающих деревьев. Данная библиотека отличается высокой скоростью работы. Модели, построенные на его основе, обладают хорошей производительностью. Поэтому он пользуется популярностью при решении задач классификации и регрессии с использованием табличных наборов данных.

**Библиотека *Scikit-learn***

*Scikit-learn* - один из наиболее широко используемых пакетов *Python* для Data Science и Machine Learning. Он позволяет выполнять множество операций и предоставляет множество алгоритмов. *Scikit-learn* поддерживает: предварительную обработку данных, уменьшение размерности, выбор модели регрессии или классификации, кластерный анализ.

**Библиотека *Pandas***

*Pandas* — программная библиотека на языке *Python* для обработки и анализа данных. Основная область применения — обеспечение работы в рамках среды *Python* не только для сбора и очистки данных, но для задач анализа и моделирования данных, без переключения на более специфичные для статобработки языки.

Пакет прежде всего предназначен для очистки и первичной оценки данных по общим показателям, например среднему значению, квантилям и другим.

**Библиотека Matplotlib**

*Matplotlib* — библиотека на языке программирования *Python* для визуализации данных двумерной и трёхмерной графикой. Пакет поддерживает многие виды графиков и диаграмм: графики, диаграммы рассеяния, столбчатые диаграммы, круговые диаграммы, диаграммы стебель-листья, контурные графики, поля градиентов, 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. Вычислительный эксперимент

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. Объектами являются женщины в возрасте от двадцати одного года индийского наследия Пима.

Признаками объектов являются:

1. Pregnancies - количество беременностей;
2. Glucose - плазменные концентрации глюкозы в крови;
3. BloodPressure - диастолическое артериальное давление;
4. SkinThickness - толщина кожи в области трицепса;
5. Insulin - количество инсулина в крови;
6. BMI - индекс массы тела;
7. DiabetesPedigreeFunction - оценка предрасположенности к диабету;
8. 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.

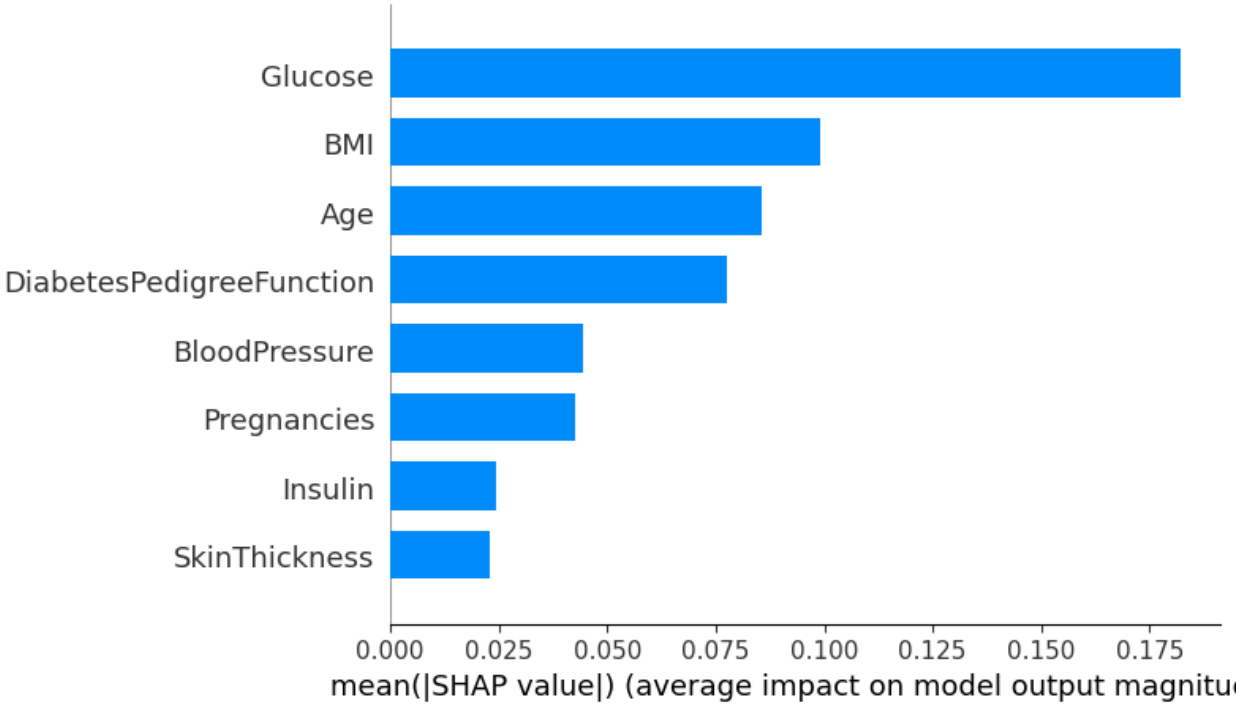


Рисунок 12. Интерпретация модели машинного обучения, построенной на основе набора данных *diabetes*, методом SHAP.

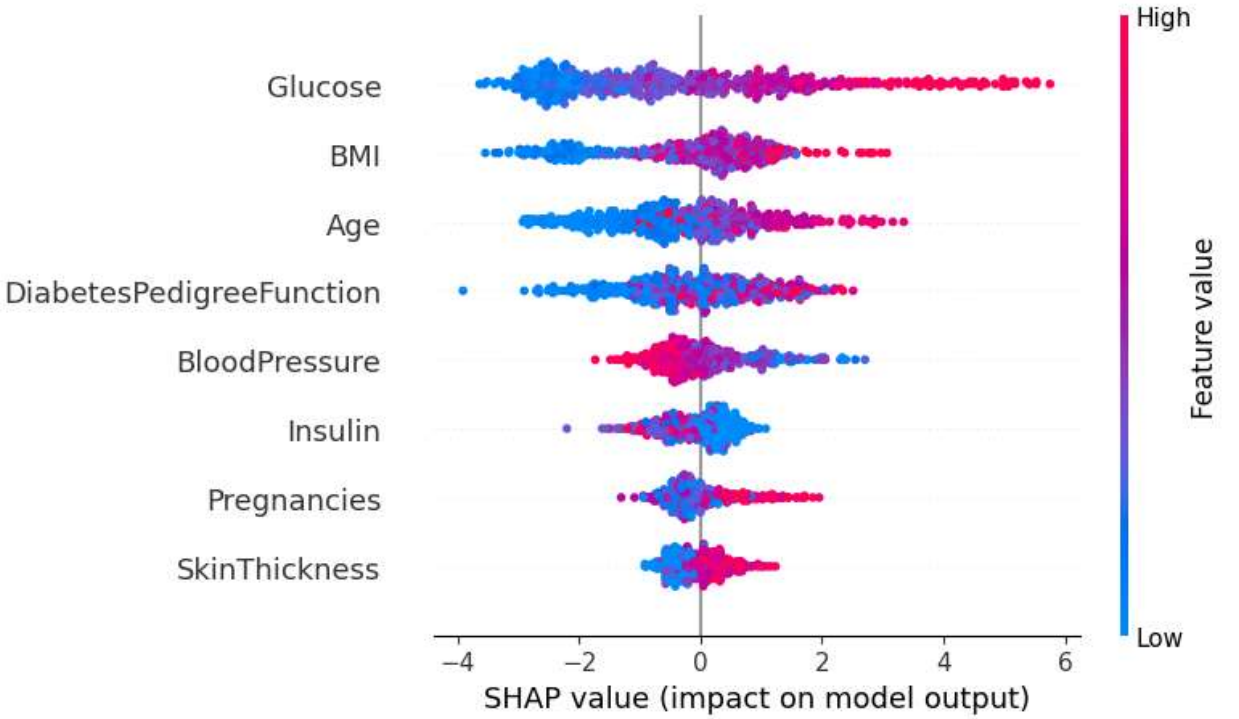


Рисунок 13. Интерпретация модели машинного обучения, построенной на основе набора данных *diabetes*, методом SHAP.

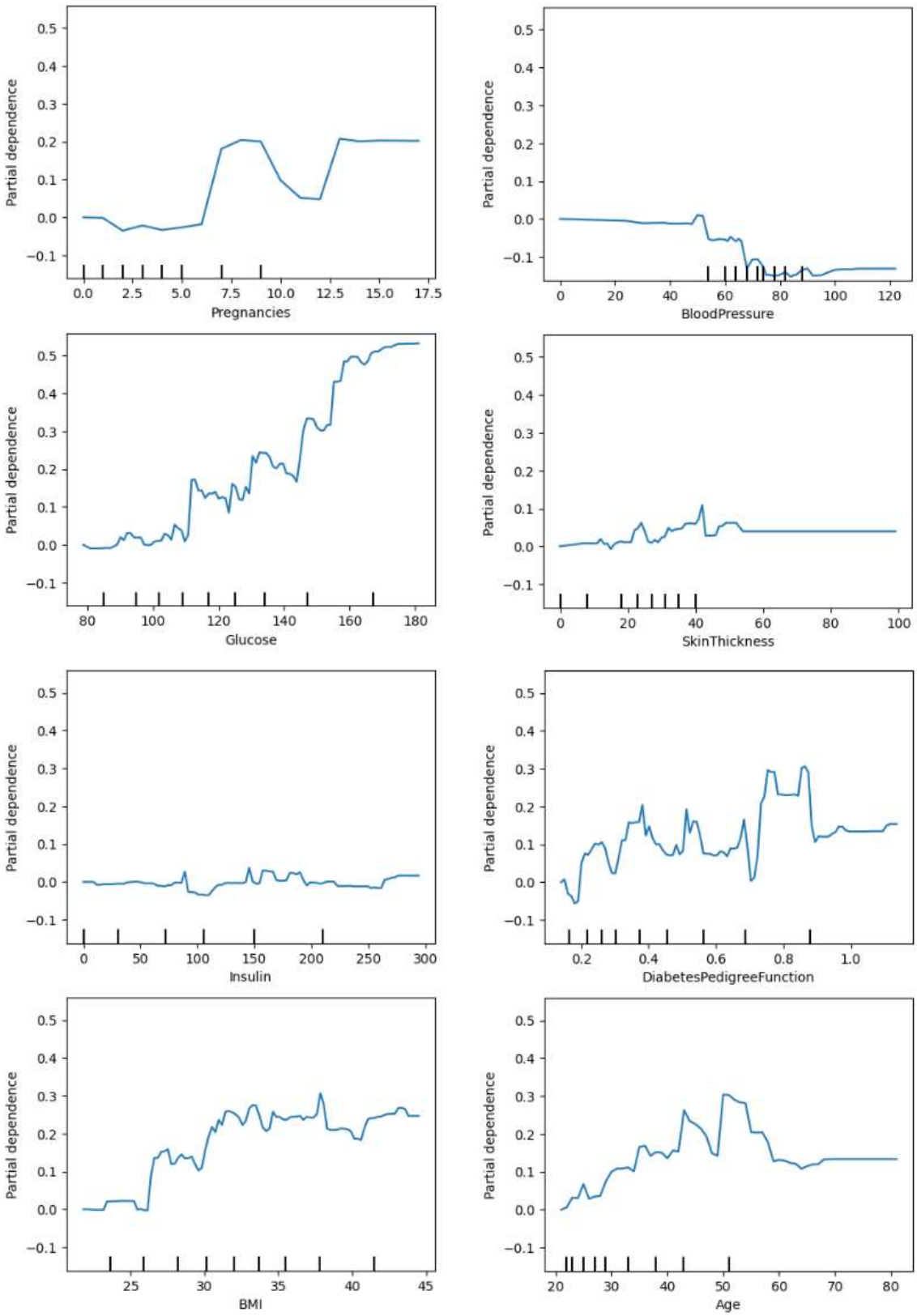


Рисунок 14. Интерпретация модели машинного обучения, построенной на основе набора данных *diabetes*, методом PDP.

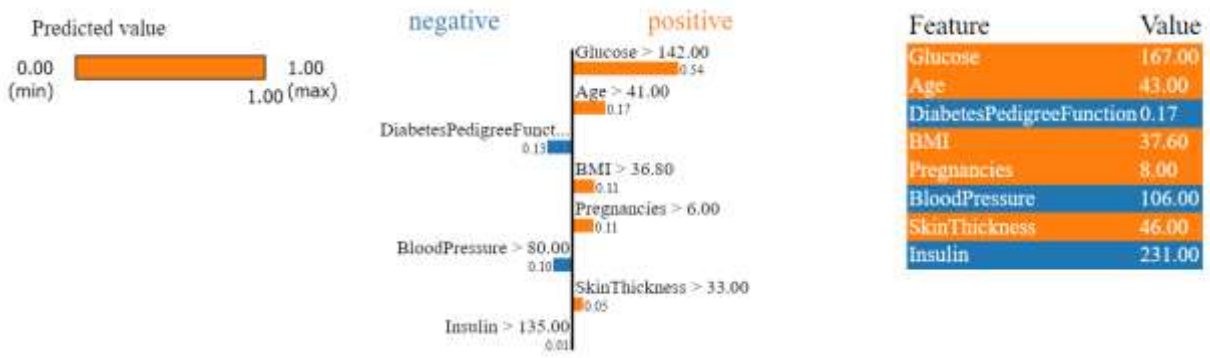


Рисунок 15. Локальная интерпретация модели машинного обучения для объекта №60, построенной на основе набора данных *diabetes*, методом LIME.

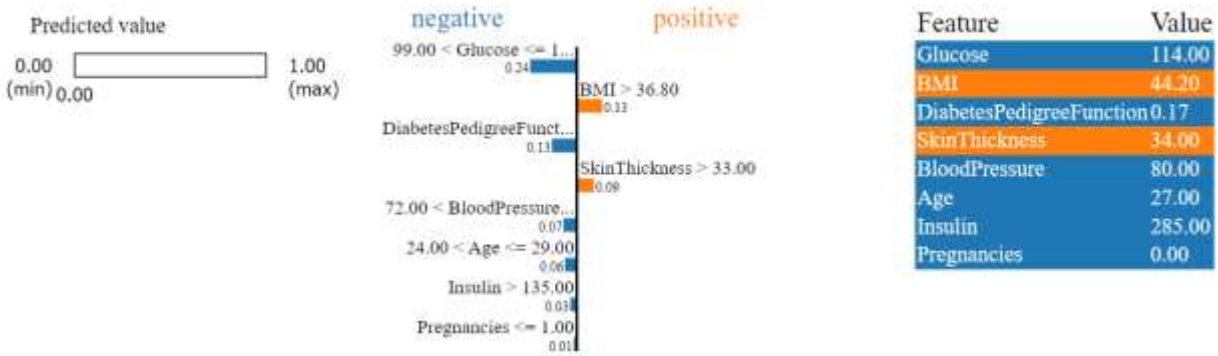


Рисунок 16. Локальная интерпретация модели машинного обучения для объекта №100, построенной на основе набора данных *diabetes*, методом LIME.

### 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. Он состоит из объектов, каждый из которых имеет восемь признаков и одну целевую переменную.

Признаками объектов являются:

1. Longitude - долгота квартала с недвижимостью;
2. Latitude - широта квартала с недвижимостью;
3. HouseAge - медиана возраста домов в квартале;
4. AveRooms - общее количество комнат в квартале;

5. AveBedrms - общее количество спален в квартале;
6. Population - население квартала;
7. AveOccup - количество семей в квартале;
8. 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.

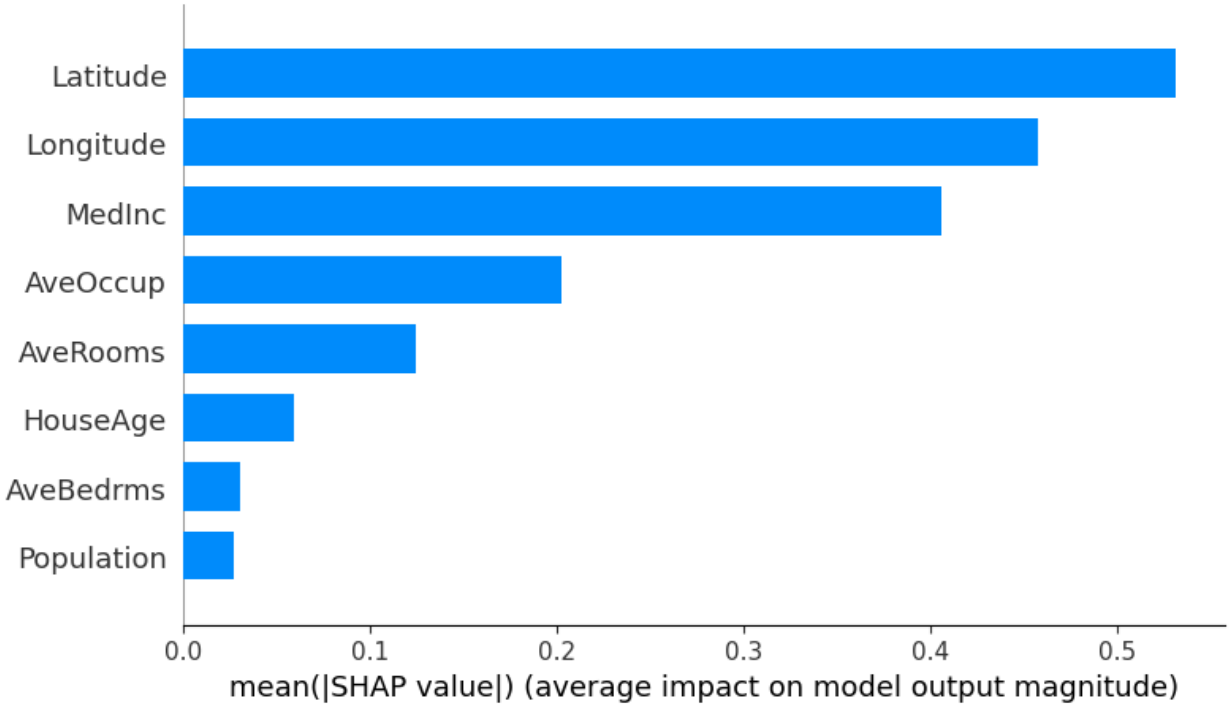


Рисунок 17. Интерпретация модели машинного обучения, построенной на основе набора данных california housing, методом SHAP.

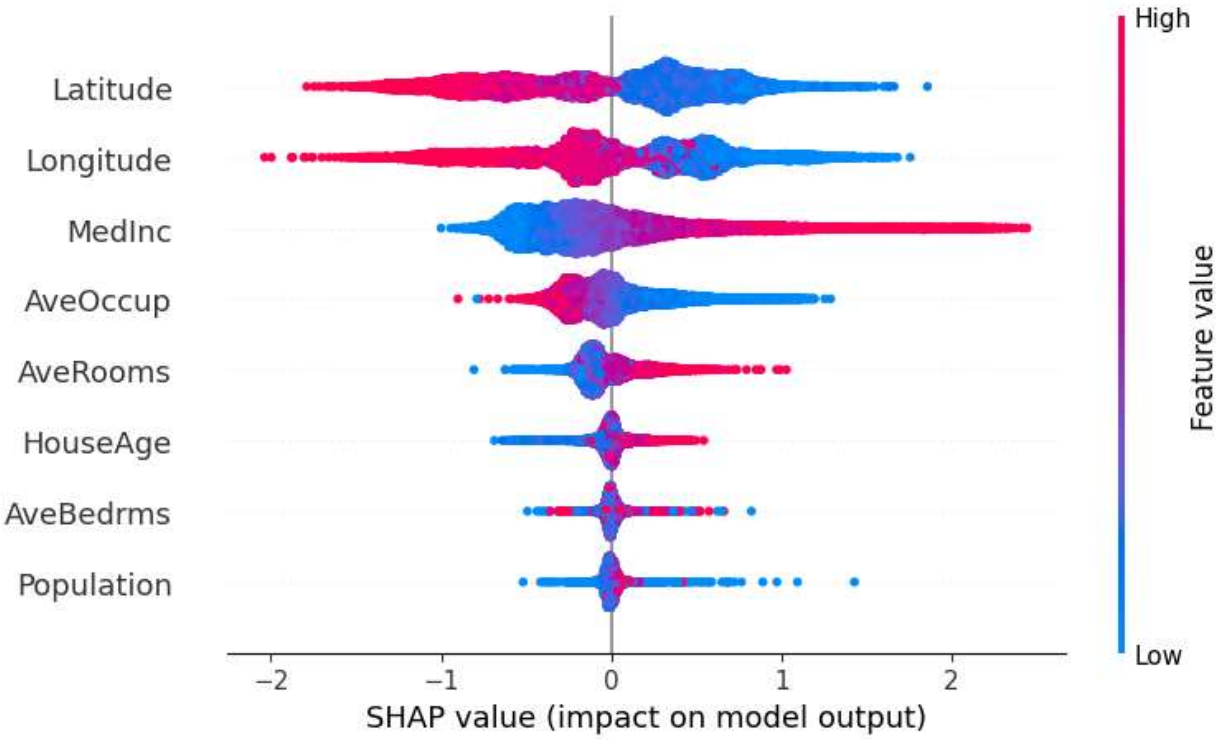


Рисунок 18. Интерпретация модели машинного обучения, построенной на основе набора данных *california housing*, методом SHAP.

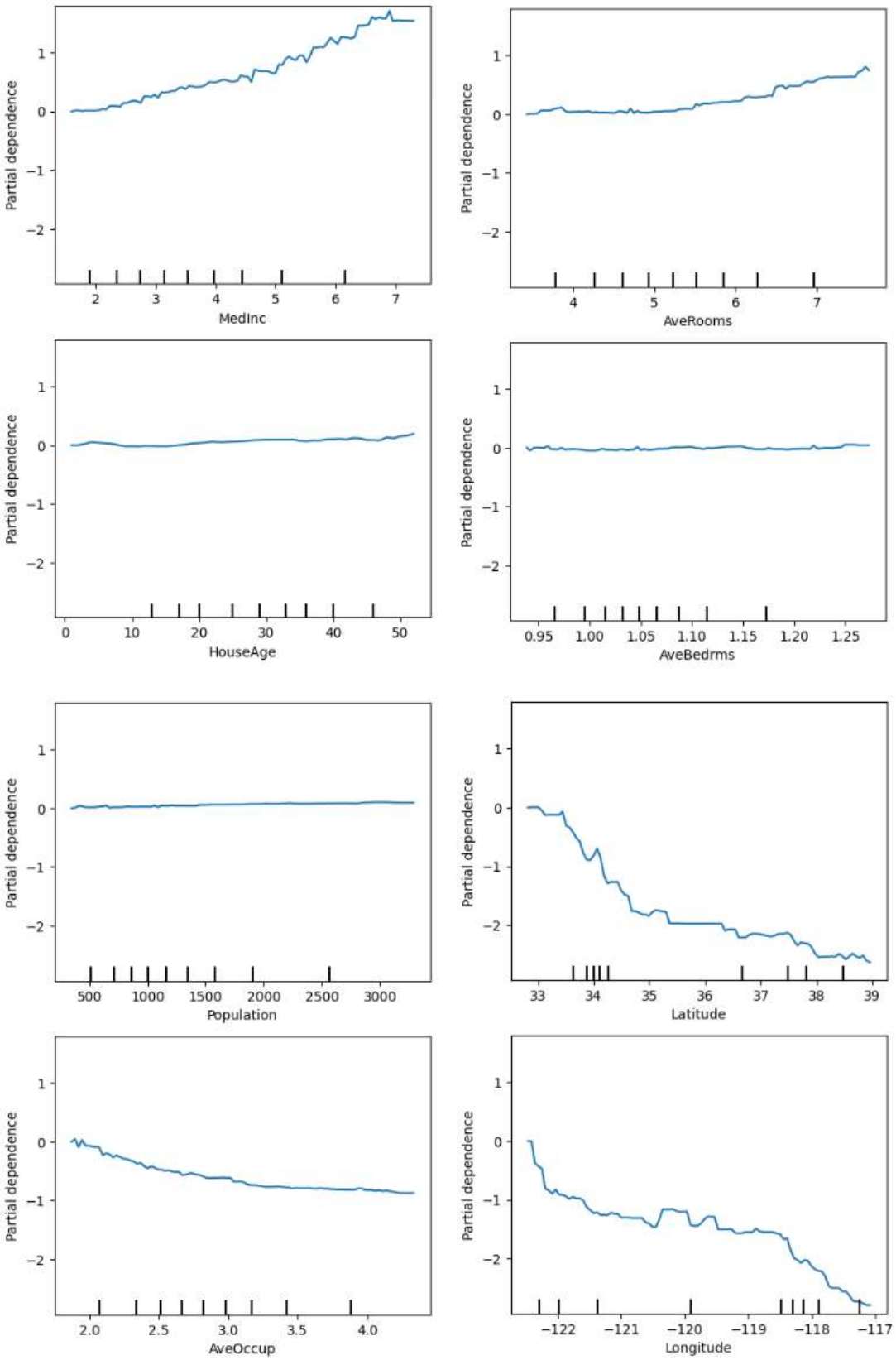


Рисунок 19. Интерпретация модели машинного обучения, построенной на основе набора данных *california housing*, методом PDP.

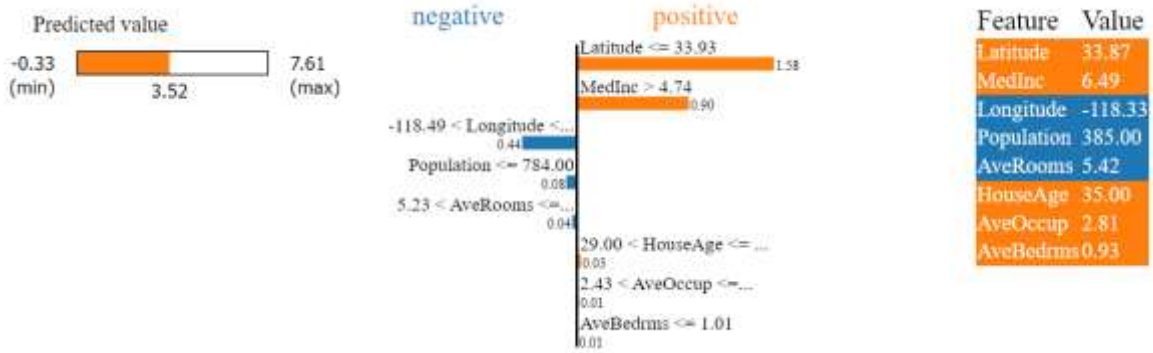


Рисунок 20. Локальная интерпретация модели машинного обучения для объекта №50, построенной на основе набора данных *california housing*, методом LIME.


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. Анализ эксперимента 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. Анализ эксперимента 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.

## 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
29. Štrumbelj E., Kononenko I. Explaining prediction models and individual predictions with feature contributions. Knowledge and Information Systems. (2013).