

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ГЛУБИННОЕ ОБУЧЕНИЕ

Направление подготовки:
09.03.03 Прикладная информатика

Направленность (профиль)
Искусственный интеллект и анализ данных

Уровень высшего образования: бакалавриат

1. Цели и задачи освоения дисциплины

Цель - приобретение студентами теоретических и практических знаний и навыков в области построения формальных математических моделей и интерпретации результатов моделирования; выработать умения по практическому применению методов машинного глубинного обучения для построения формальных математических моделей и интерпретации результатов моделирования при решении прикладных задач в различных прикладных областях и использования различных программных инструментов анализа баз данных и систем машинного обучения.

Для достижения поставленной цели при изучении дисциплины решаются следующие задачи:

- знакомство обучаемых с основными понятиями, терминологией и областями применения машинного глубинного обучения и глубокого обучения многослойных нейронных сетей;
- формирование у обучаемых способности ставить задачи машинного и глубинного обучения при обработке больших объемов данных;
- формирование у обучаемых способности оценивать эффективность применения методов машинного обучения при решении различных прикладных задач;
- формирование у обучаемых способности проводить вычислительную оптимизацию методов глубинного обучения искусственных нейронных сетей;
- изучение особенностей решения задач на обучение нейросетей с подкреплением и используемых для этого методов;
- освоение навыков применения методов машинного обучения и глубинного обучения нейронных сетей на практике при разработке блоков анализа больших данных в современных промышленных системах.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Глубинное обучение» относится к части, формируемой участниками образовательного процесса, учебного плана ОПОП ВО по направлению подготовки 09.03.03 Прикладная информатика.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

3.1. Компетенции обучающегося, формируемые в результате освоения дисциплины с текстом:

Процесс изучения дисциплины направлен на формирование у обучающихся следующих компетенций (результатов освоения образовательной программы):

| Коды компетенций | Содержание компетенций |
|------------------|--|
| ПК-7 | Способность выполнять интеллектуальный анализ больших данных |
| ПК-9 | Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных |

3.2. Взаимосвязь планируемых результатов обучения по дисциплине с формируемыми компетенциями образовательной программы

| Коды и формулировка компетенции | Индикаторы компетенции | Запланированные результаты обучения |
|--|---|--|
| ПК-7 Способность выполнять интеллектуальный анализ больших | ПК 7.1 Знать методы и инструментальные средства интеллектуального | ПК 7.1.1 знать методы и инструментальные средства для машинного обучения и анализа больших данных - предметную область анализа больших данных - теоретические и прикладные основы анализа данных |

| | | |
|---|---|---|
| данных | <p>анализа больших данных ПК 7.2 Уметь выбирать средства представления результатов аналитики больших данных ПК 7.3 Владеть техническими, программными средствами для разработки алгоритмов и программ в области интеллектуального анализа данных</p> | <p>ПК 7.1.2 уметь использовать инструментальные средства машинного обучения и анализа больших данных - проводить сравнительный анализ методов и инструментальных средств анализа больших данных ПК 7.1.3 владеть инструментальными средствами машинного обучения и анализа больших данных ПК 7.2.1 знать средства визуализации результатов решения задач машинного обучения и анализа больших данных методы интерпретации и визуализации решения задач машинного обучения и анализа больших данных ПК 7.2.2 уметь использовать стандартные программные библиотеки для визуализации решений задач машинного обучения и анализа данных ПК 7.2.3 владеть стандартными программными библиотеками для решения задач машинного обучения и анализа данных ПК 7.3.1 знать программные платформы и библиотеки для решения задач машинного обучения и анализа больших данных стандарты проведения анализа данных ПК 7.3.2 уметь использовать программные платформы и библиотеки для решения задач машинного обучения и анализа больших данных ПК 7.3.3 владеть программными средствами для разработки алгоритмов машинного обучения, алгоритмами построения искусственных нейронных сетей</p> |
| ПК-9 Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных | <p>ПК 9.1 Знать принципы решения задач машинного обучения и интеллектуального анализа данных ПК 9.2 Уметь создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных ПК 9.3 Владеть навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных</p> | <p>ПК 9.1.1 знать основные принципы решения задач машинного обучения и анализа данных ПК 9.1.2 уметь проводить спецификацию задачи, реализовывать программы на алгоритмических языках высокого уровня, интерпретировать полученные результаты ПК 9.1.3 владеть методами и алгоритмами машинного обучения ПК 9.2.1 знать математические и алгоритмические модели интеллектуального анализа данных, методы оценки временных и стоимостных характеристик машинного обучения и анализа данных ПК 9.2.2 уметь создавать алгоритмические модели типовых прикладных задач обработки информации, реализовывать программы на алгоритмических языках высокого уровня ПК 9.2.3 владеть навыками использования прикладных библиотек для решения задач машинного обучения ПК 9.3.1 описательные и прогнозные аналитические модели для интеллектуального анализа данных - использовать современные программные инструменты интеллектуального анализа данных 9.3.2 разрабатывать и оценивать модели больших данных 9.3.3 навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных, навыками выявления требований к результатам анализа больших данных в предметной области</p> |

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 часов.

4.1. Объем дисциплины

| | |
|--------------------|-------------|
| Вид учебной работы | Всего часов |
|--------------------|-------------|

| | Очная форма обучения | Очно-заочная форма обучения | Заочная форма обучения |
|---|----------------------|-----------------------------|------------------------|
| Общая трудоемкость дисциплины | 144 | не реализуется | не реализуется |
| Контактная работа обучающегося с преподавателем (по видам учебных занятий) (всего), в том числе: | 32 | | |
| <i>Занятия лекционного типа</i> | 16 | | |
| <i>Занятия семинарского типа (практич., семин., лаборат. и др.)</i> | 16 | | |
| <i>Самостоятельная работа под руководством преподавателя</i> | 76 | | |
| Самостоятельная работа обучающихся СРС/подготовка к экзамену (зачету) в соответствии с учебным планом | 36 | | |
| Вид промежуточной аттестации (зачет/экзамен) | экзамен | | |

4.2. Разделы и темы дисциплины, их трудоемкость по видам учебных занятий

4.2.1. Темы дисциплины, их трудоемкость по видам учебных занятий для очной формы обучения:

Не реализуется

4.2.2. Темы дисциплины, их трудоемкость по видам учебных занятий для очно-заочной формы обучения:

| № п/п | Наименование разделов, тем учебных занятий | Всего часов | Контактная работа обучающегося с преподавателем (по видам учебных занятий) (всего), в том числе: | | | | Процедура оценивания/оцениваемые компетенции |
|-------|--|-------------|--|----|----|-----|--|
| | | | ЛЗ | СЗ | ЛР | СРС | |
| 1 | Наука о данных | 12 | 2 | | | 10 | Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9 |
| 2 | Задачи обучения | 12 | 2 | | | 10 | Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9 |
| 3 | Математические основы машинного обучения | 12 | 2 | 6 | | 4 | Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9 |
| 4 | Алгоритмы и методы машинного обучения | 18 | 2 | 4 | | 12 | Тестирование, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9 |
| 5 | Деревья решений | 12 | 2 | 2 | | 8 | Участие в устном опросе, решение задач, выполнение заданий и упражнений для |

| | | | | | | | |
|---|---|-----|----|----|--|----|--|
| | | | | | | | самостоятельной работы / ПК-7, ПК9 |
| 6 | Контроль качества и ансамбли алгоритмов | 12 | 2 | 2 | | 8 | Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9 |
| 7 | Введение в рекомендательные системы | 12 | 2 | 2 | | 8 | Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9 |
| 8 | Сложность алгоритмов | 18 | 2 | | | 16 | Тестирование, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9 |
| | Экзамен | 36 | | | | 36 | Сдача экзамен |
| | Итого | 144 | 16 | 16 | | 76 | |

4.2.3. Темы дисциплины, их трудоемкость по видам учебных занятий для заочной формы обучения:

Не реализуется

4.3. Содержание дисциплины

Тема 1 Наука о данных

Python: инструментарий. Библиотека для матричных вычислений и линейной алгебры NumPy. Библиотека для научных вычислений SciPy. Библиотека для визуализации Matplotlib. Библиотека для машинного обучения Scikit-Learn. Библиотека для обработки данных Pandas. Наука о данных (Data Science). Статистика (Statistics).

Искусственный интеллект (Artificial Intelligence). Анализ данных (Data Mining). Машинное обучение (Machine learning). Большие данные (Big Data).

Обучение с учителем (с размеченными данными / метками), целевая функция, объект, метка, классификация.

Прогнозирование. Пространство объектов признаковое пространство. Извлечение признаков.

Тема 2 Задачи обучения

Задачи обучения. Визуализация задач. Схема решения задачи машинного обучения. Схема проверки алгоритма. Подход основанный на близости. Параметрические модели. Переобучение. Обучение без учителя (unsupervised Learning) с неразмеченными данными, без меток. Обучение с частично размеченными данными (Semi-Supervised Learning). Трансдуктивное обучение (transductive learning). Обучение с подкреплением (Reinforcement Learning).

Структурный вывод (Structured output). Активное обучение (Active Learning). Онлайн-обучение (Online Learning).

Обучение (выучивание) признаков (Feature Learning). Сложности в ML. Примеры модельных задач.

Тема 3 Математические основы машинного обучения

Бритва Оккама. Теорема о бесплатном сыре. Футбольный оракул. Элементы теории вероятностей и математической статистики. Функция распределения. Задание распределений. Средние и отклонения. Математическое ожидание. Дисперсия. Условная плотность, маргинализация и обуславливание. Смещение. Несмещённая (unbiased) оценка. Асимптотически несмещённая оценка.

Состоятельность (Consistency). Несмещённые оценки для нормального распределения. Оценка максимального правдоподобия (ММП / MLE). Дивергенция Кульбака-Лейблера ковариация и корреляция. Оценка плотности гистограммного подхода. Парзеновский подход

.Нормальное распределение. Центральная предельная теорема. Теория информации. Проклятие размерности. Сингулярное разложение матрицы (SVD) матричное дифференцирование. Методы безусловной оптимизации. Методы нулевого порядка. Методы первого порядка. Методы второго порядка. Градиент. Антиградиент. Градиентный спуск. Проблема постоянного шага. Проблема масштаба признаков. Наискорейший градиентный спуск. Стохастический градиентный спуск. Обучение: Пакетное, онлайн, по минибатчам. Метод градиентного спуска в машинном обучении. Пример градиентного спуска для оптимизации квадратичного функционала. Стационарные точки. Метод Ньютона. Квазиньютоновские методы. Оптимизация с ограничениями.

Тема 4 Алгоритмы и методы машинного обучения

Метрические алгоритмы (distance-based). Ближайший центроид (Nearest centroid algorithm). Подход, основанный на близости kNN в задаче классификации kNN в задаче регрессии. Обоснование 1NN Ленивые (Lazy) и нетерпеливые (Eager) алгоритмы. Весовые обобщения kNN. Различные метрики: Минковского, Евклидова, Манхэттенская, Махаланобиса, Canberra distance, Хэмминга, косинусное, расстояние Джаккарда, DTW, Левенштейна.

Приложения метрического подхода: нечёткий матчнинг таблиц, Ленкор, в DL, классификация текстов. Эффективные методы поиска ближайших соседей РегрессияНадарая-Ватсона.

Линейная регрессия. Обобщённая линейная регрессия. Проблема вырожденности матрицы. Регуляризация.

Основные виды регуляризации. Гребневая регрессия (Ridge Regression). LASSO (Least Absolute Selection and Shrinkage Operator). Elastic Net. Селекция признаков. Ошибка с весами. Устойчивая регрессия (Robust Regression).

Линейные скоринговые модели в задаче бинарной классификации. Логистическая регрессия. Probit-регрессия. Многоклассовая логистическая регрессия. Линейный классификатор. Перцептрон. Оценка функции ошибок через гладкую функцию

Тема 5 Деревья решений

Деревья решений (CART). Предикаты / ветвления. Ответы дерева. Критерии расщепления в задачах классификации: Missclassification criteria, энтропийный, Джини. Критерии остановки при построении деревьев. Проблема переобучения для деревьев. Подрезка (post-pruning).

Классические алгоритмы построения деревьев решений: ID3, C5.0. Важности признаков. Проблема пропусков (Missing Values). Категориальные признаки. Сравнение: деревья и линейные модели

Случайный лес, его параметры, их настройка. Бэггинг и OOB (out of bag). Важность признаков. Близость (Proximity) с помощью RF. Extreme Random Trees. Градиентный бустинг над деревьями, его параметры, современные реализации, Продвинутое методы оптимизации

Тема 6 Контроль качества и ансамбли алгоритмов

Проблема контроля качества. Выбора модели (Model Selection) в широком смысле. Правила разбиения выборки. Отложенный контроль (held-out data, hold-out set).

Скользкий контроль (cross-validation). Бутстреп (bootstrap). Контроль по времени (out-of-time-контроль). Локальный контроль. Кривые обучения (Learning Curves). Перебор параметров

Ансамбли алгоритмов: примеры и обоснование комитеты (голосование) / усреднение. Бэггинг Кодировки / перекодировки ответов, ECOC. Стекинг и блендинг Бустинг: AdaBoost, Forward stagewise additive modeling (FSAM) «Ручные методы». Однородные ансамбли

Тема 7 Введение в рекомендательные системы

Введение в рекомендательные системы. Персонализация, онлайн и оффлайн рекомендации. Рекомендация по контенту (content based methods). Коллаборативная фильтрация: GroupLensалгоритм, SVD, SVD++, timeSVD++, адаптация SVD под социальные связи. One-class recommendation.

Факторизационная машина, факторизационная машина с полями (FFM – field-aware factorization machine). Библиотека libFM. Простые методы рекомендаций: техника FPM – Frequent Pattern Mining. Deep Semantic Similarity. Model (DSSM). Контекст рекомендации. Knowledge-based Recommendations. Важность объяснений (explanations).

Использование дополнительной информации. Современные тренды в практике построения рекомендательных систем.

Тема 8 Сложность алгоритмов

Переобучение (overfitting). Недообучение (underfitting). Сложность (complexity) модели алгоритмов. Разброс (variance). Способность модели алгоритмов настраиваться на целевую зависимость. Способы борьбы с переобучением: использование выборки специальной структуры; увеличение объёма данных; «аугментация»; «улучшение качества данных»; регуляризация; уменьшение избыточности данных

4.4. Темы семинарских занятий и лабораторных работ

Тема 1 Наука о данных

Решение задач по теме 1.

Тема 2 Задачи обучения

Решение задач по теме 2.

Тема 3 Модульное тестирование

Решение задач по теме 3.

Тема 4 Алгоритмы и методы машинного обучения

Решение задач по теме 4. Тестирование по темам 1,2,3 и 4.

Тема 5 Деревья решений

Решение задач по теме 5.

Тема 6 Контроль качества и ансамбли алгоритмов

Решение задач по теме 6.

Тема 7 Введение в рекомендательные системы

Решение задач по теме 7.

Тема 8 Сложность алгоритмов

Решение задач по теме 8. Тестирование по темам 1, 2, 3,4, 5, 6, 7 и 8

5. Учебно-методическое обеспечение самостоятельной работы

5.1. Задания для самостоятельной подготовки к занятиям семинарского типа

Семинарское занятие №1.

Тема занятия: Элементы теории вероятностей и математической статистики

Задания (вопросы) для подготовки:

Задание распределений.

Средние и отклонения.

Условная плотность, маргинализация и обуславливание.

Точечное оценивание.

Оценка максимального правдоподобия.

Дивергенция Кульбака-Лейблера ковариация и корреляция.

Оценка плотности гистограммного подхода.

Парзеновский подход.

Нормальное распределение.

Центральная предельная теорема.

Теория информации.

Проклятие размерности.

Сингулярное разложение матрицы (SVD) матричное дифференцирование.

Семинарское занятие 2.

Тема занятия: Метод градиентного спуска в машинном обучении описание алгоритма:

Рабочая область функции (заданный интервал) разбита на несколько точек. Выбраны точки локальных минимумов. После этого все координаты передаются функции в качестве аргументов и выбирается аргумент, дающий наименьшее значение. Затем применяется метод градиентного спуска.

реализация:

импортируем библиотеки и определяем константы

```
import numpy as np
import matplotlib.pyplot as plot

radius = 8 # working plane radius
centre = (global_epsilon, global_epsilon) # centre of the working circle
arr_shape = 100 # number of points processed / 360
step = radius / arr_shape # step between two points
```

Функция, для которой рассчитывается минимум:

```
def differentiable_function(x, y):
    return np.sin(x) * np.exp((1 - np.cos(y)) ** 2) + np.cos(y) * np.exp((1 - np.sin(x)) ** 2) + (x - y) ** 2
```

Приращение аргумента

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{\Delta f(x)}{\Delta x}$$

Так как предел аргумента стремится к нулю, точность должна быть небольшой по сравнению с радиусом рабочей плоскости:

```
global_epsilon = 0.000000001 # argument increment for derivative
```

поворот вектора:

$$R\mathbf{v} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \cos \theta - y \sin \theta \\ x \sin \theta + y \cos \theta \end{bmatrix}$$

Если вращение применяется к вектору $(x, 0)$, повернутый вектор будет вычисляться следующим образом:

```
def rotate_vector(length, a):
    return length * np.cos(a), length * np.sin(a)
```

Вычисление производной по оси X, где эpsilon - значение x:

```
def derivative_x(epsilon, arg):
```

```

return (differentiable_function(global_epsilon + epsilon, arg) -
differentiable_function(epsilon, arg)) / global_epsilon

```

Расчет градиента:

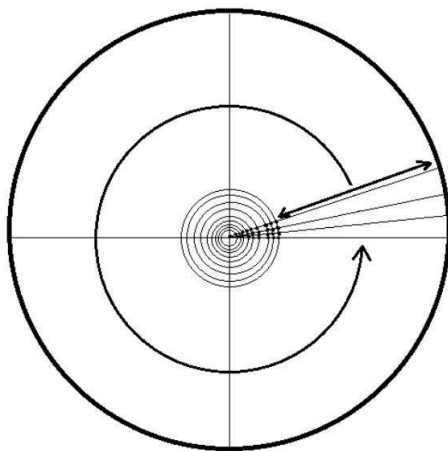
$$\nabla f = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} + \frac{\partial f}{\partial z} \mathbf{k}$$

Поскольку градиент вычисляется для 2D-функции, k равно нулю

```

gradient = derivative_x(x, y) + derivative_y(y, x)

```



360 * arr_shape dots

Возвращаемое значение представляет собой массив приблизительных локальных минимумов.

```

def calculate_flip_points():
    flip_points = np.array([0, 0])
    points = np.zeros((360, arr_shape), dtype=bool)
    cx, cy = centre

    for i in range(arr_shape):
        for alpha in range(360):
            x, y = rotate_vector(step, alpha)
            x = x * i + cx
            y = y * i + cy
            points[alpha][i] = derivative_x(x, y) + derivative_y(y, x) > 0
            if not points[alpha][i - 1] and points[alpha][i]:
                flip_points = np.vstack((flip_points, np.array([alpha, i -
1])))

    return flip_points

```

Выбор точки из flip_points, значение функции от которой минимально:

```

def pick_estimates(positions):
    vx, vy = rotate_vector(step, positions[1][0])
    cx, cy = centre
    best_x, best_y = cx + vx * positions[1][1], cy + vy * positions[1][1]

    for index in range(2, len(positions)):
        vx, vy = rotate_vector(step, positions[index][0])
        x, y = cx + vx * positions[index][1], cy + vy * positions[index][1]
        if differentiable_function(best_x, best_y) >
differentiable_function(x, y):
            best_x = x
            best_y = y

    for index in range(360):
        vx, vy = rotate_vector(step, index)
        x, y = cx + vx * (arr_shape - 1), cy + vy * (arr_shape - 1)
        if differentiable_function(best_x, best_y) >
differentiable_function(x, y):
            best_x = x
            best_y = y

    return best_x, best_y

```

Метод градиентного спуска:

```

def gradient_descent(best_estimates, is_x):
    derivative = derivative_x if is_x else derivative_y
    best_x, best_y = best_estimates
    descent_step = step
    value = derivative(best_y, best_x)

    while abs(value) > global_epsilon:
        descent_step *= 0.95
        best_y = best_y - descent_step \
            if derivative(best_y, best_x) > 0 else best_y + descent_step
        value = derivative(best_y, best_x)

    return best_y, best_x

```

Нахождение точки минимума:

```

def find_minimum():
    return
gradient_descent(gradient_descent(pick_estimates(
calculate_flip_points()),
False), True)

```

Формирование сетки точек для построения:

```

def get_grid(grid_step):
    samples = np.arange(-radius, radius, grid_step)
    x, y = np.meshgrid(samples, samples)

    return x, y, differentiable_function(x, y)

```

Построение графика:

```
def draw_chart(point, grid):
    point_x, point_y, point_z = point
    grid_x, grid_y, grid_z = grid
    plot.rcParams.update({
        'figure.figsize': (4, 4),
        'figure.dpi': 200,
        'xtick.labelsize': 4,
        'ytick.labelsize': 4
    })
    ax = plot.figure().add_subplot(111, projection='3d')
    ax.scatter(point_x, point_y, point_z, color='red')
    ax.plot_surface(grid_x, grid_y, grid_z, rstride=5, cstride=5, alpha=0.7)

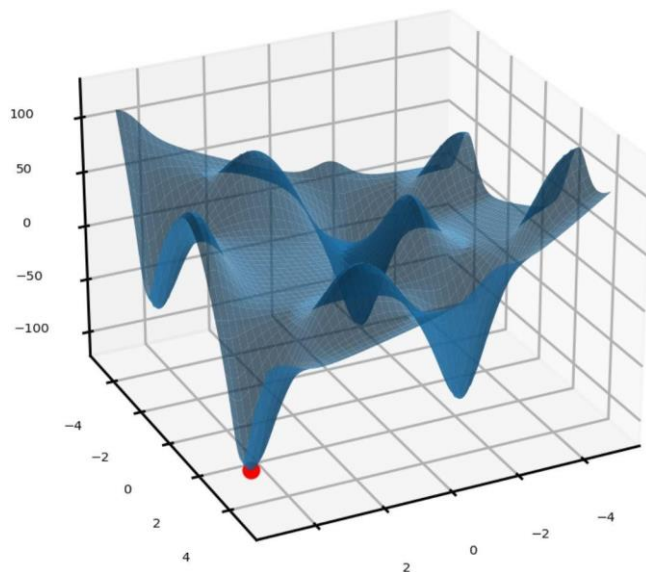
plot.show()
```

Функция main:

```
if __name__ == '__main__':
    min_x, min_y = find_minimum()
    minimum = (min_x, min_y, differentiable_function(min_x, min_y))

draw_chart(minimum, get_grid(0.05))
```

График:



Исходный код:

```
import numpy as np
import matplotlib.pyplot as plot

radius = 8 # working plane radius
global_epsilon = 0.000000001 # argument increment for
derivative
```

```

centre = (global_epsilon, global_epsilon) # centre of the working circle
arr_shape = 100 # number of points processed /
360
step = radius / arr_shape # step between two points

def differentiable_function(x, y):
    return np.sin(x) * np.exp((1 - np.cos(y)) ** 2) + \
           np.cos(y) * np.exp((1 - np.sin(x)) ** 2) + (x - y) ** 2

def rotate_vector(length, a):
    return length * np.cos(a), length * np.sin(a)

def derivative_x(epsilon, arg):
    return (differentiable_function(global_epsilon + epsilon, arg) -
            differentiable_function(epsilon, arg)) / global_epsilon

def derivative_y(epsilon, arg):
    return (differentiable_function(arg, epsilon + global_epsilon) -
            differentiable_function(arg, epsilon)) / global_epsilon

def calculate_flip_points():
    flip_points = np.array([0, 0])
    points = np.zeros((360, arr_shape), dtype=bool)
    cx, cy = centre

    for i in range(arr_shape):
        for alpha in range(360):
            x, y = rotate_vector(step, alpha)
            x = x * i + cx
            y = y * i + cy
            points[alpha][i] = derivative_x(x, y) + derivative_y(y, x) > 0
            if not points[alpha][i - 1] and points[alpha][i]:
                flip_points = np.vstack((flip_points, np.array([alpha, i -
1])))

    return flip_points

def pick_estimates(positions):
    vx, vy = rotate_vector(step, positions[1][0])
    cx, cy = centre
    best_x, best_y = cx + vx * positions[1][1], cy + vy * positions[1][1]

    for index in range(2, len(positions)):
        vx, vy = rotate_vector(step, positions[index][0])
        x, y = cx + vx * positions[index][1], cy + vy * positions[index][1]
        if differentiable_function(best_x, best_y) >
differentiable_function(x, y):
            best_x = x
            best_y = y

    for index in range(360):
        vx, vy = rotate_vector(step, index)
        x, y = cx + vx * (arr_shape - 1), cy + vy * (arr_shape - 1)
        if differentiable_function(best_x, best_y) >
differentiable_function(x, y):
            best_x = x
            best_y = y

```

```

return best_x, best_y

def gradient_descent(best_estimates, is_x):
    derivative = derivative_x if is_x else derivative_y
    best_x, best_y = best_estimates
    descent_step = step
    value = derivative(best_y, best_x)

    while abs(value) > global_epsilon:
        descent_step *= 0.95
        best_y = best_y - descent_step \
            if derivative(best_y, best_x) > 0 else best_y + descent_step
        value = derivative(best_y, best_x)
    return best_y, best_x

def find_minimum():
    return
    gradient_descent(gradient_descent(pick_estimates(
        calculate_flip_points()), False), True)

def get_grid(grid_step):
    samples = np.arange(-radius, radius, grid_step)
    x, y = np.meshgrid(samples, samples)
    return x, y, differentiable_function(x, y)

def draw_chart(point, grid):
    point_x, point_y, point_z = point
    grid_x, grid_y, grid_z = grid
    plot.rcParams.update({
        'figure.figsize': (4, 4),
        'figure.dpi': 200,
        'xtick.labelsize': 4,
        'ytick.labelsize': 4
    })
    ax = plot.figure().add_subplot(111, projection='3d')
    ax.scatter(point_x, point_y, point_z, color='red')
    ax.plot_surface(grid_x, grid_y, grid_z, rstride=5, cstride=5, alpha=0.7)
    plot.show()

if __name__ == '__main__':
    min_x, min_y = find_minimum()
    minimum = (min_x, min_y, differentiable_function(min_x, min_y))
    draw_chart(minimum, get_grid(0.05))

```

Семинарское занятие 3.

Тема занятия: Метрические алгоритмы

Задания (вопросы) для подготовки:

1. Ближайший центроид (Nearest centroid algorithm).
2. Подход, основанный на близости kNN в задаче классификации kNN в задаче регрессии.
3. Обоснование 1NN Ленивые (Lazy) и нетерпеливые (Eager) алгоритмы.
4. Весовые обобщения kNN.
5. Различные метрики: Минковского, Евклидова, Манхэттенская, Махаланобиса, Canberra distance, Хэмминга, косинусное, расстояние Джаккарда, DTW, Левенштейна.

6. Приложения метрического подхода: нечёткий матчинг таблиц, Ленкор, в DL, классификация текстов.

Семинарское занятие 4.

Тема занятия: Однослойный перцептрон на языке Python

Задания (вопросы) для подготовки:

Реализация алгоритма перцептона на python

Стохастический градиентный спуск Наборы данных (Iris)

Предварительная обработка

Построение прогнозов

Весы тренировочные сети

Моделирование набора данных

```
import random

def train(X, y, syn0, lens0, lens01, lr):
    errorreturn = 0
    for i in range(0, lens0):
        a = 0
        for j in range(0, lens01):
            a = a + syn0[i][j] * X[j]
        b = 1 / (1 + 2.718281828459045235360287471352662497757 ** -a)
        error = y[i] - b
        errorreturn = errorreturn + abs(error)
        delta = (error) * 1 / (1 + 2.718281828459045235360287471352662497757
** -b) * lr
        for z in range(0, lens01):
            syn0[i][z] = syn0[i][z] + X[z] * delta
    return syn0, errorreturn

def predict(syn0, X, lens0, lens01):
    ret = []
    for i in range(0, lens0):
        a = 0
        for j in range(0, lens01):
            a = a + syn0[i][j] * X[j]
        b = 1 / (1 + 2.718281828459045235360287471352662497757 ** -a)
        ret.append(b)
    return ret

def trainmass(X, y, syn, lens0, lens01, lr, iter):
    for i in range(iter):
        for j in range(lens01):
            syn, er = train(X[j], y[j], syn, lens0, lens01, lr)
    return syn, er

def create(x, y):
    syn0 = []
    for z in range(0, x):
        h = []
        for i in range(0, y):
            h.append(random.uniform(-0.1, 0.1))
        syn0.append(h)
    return syn0

syn = create(2, 4)
X = [[0.1, 0.2, 0.3, 0.4], [0.2, 0.3, 0.4, 0.5], [0.4, 0.5, 0.6,
0.7], [0.5, 0.6, 0.7, 0.8]]
```

```

y = [[0.5, 0.6],[0.6, 0.7],[0.8, 0.9],[0.9, 1]]
g, r = trainmass(X, y, syn, 2, 4, 1, 60000)
h = predict(syn, X[0], 2, 4)
h2 = predict(syn, [0.5, 0.6, 0.7, 0.8], 2, 4)
print(h)
print(h2)

```

Семинарское занятие 5.

Тема занятия: Деревья решений

Задания (вопросы) для подготовки:

1. Предикаты / ветвления.
2. Ответы дерева.
3. Критерии расщепления в задачах классификации: Missclassification criteria, энтропийный, Джини.
4. Критерии остановки при построении деревьев.
5. Проблема переобучения для деревьев.
6. Подрезка (post-pruning).
7. Классические алгоритмы построения деревьев решений: ID3, C5.0.
8. Сравнение: деревья vs линейные модели применения дерева решений из библиотеки Scikit-learn для синтетических данных

```

%pylab inline
import numpy as np
import pylab as plt
plt.rcParams['figure.figsize'] = (10.0, 8.0)

```

Сгенерируем данные. Два класса будут сгенерированы из двух нормальных распределений с разными средними.

```

# первый класс
train_data = np.random.normal(size=(100, 2))
train_labels = np.zeros(100)

# добавляем второй класс
train_data = np.r_[train_data, np.random.normal(size=(100, 2), loc=2)]
train_labels = np.r_[train_labels, np.ones(100)]

```

Вспомогательная функция, которая будет возвращать решетку для дальнейшей визуализации

```

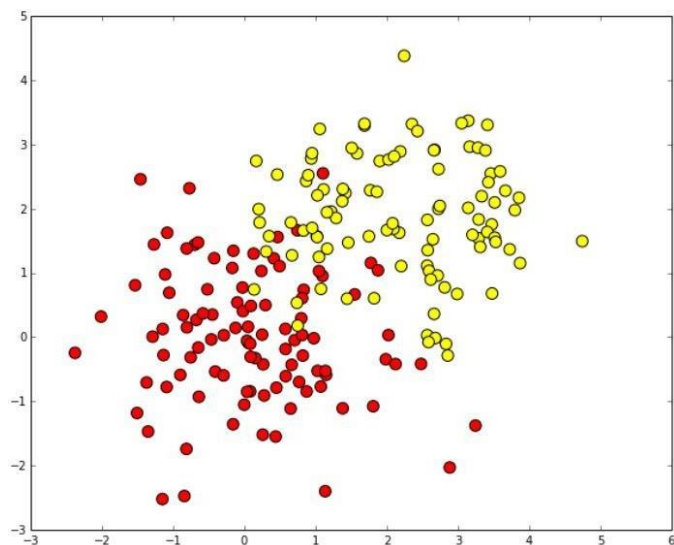
def get_grid(data):
    x_min, x_max = data[:, 0].min() - 1, data[:, 0].max() + 1
    y_min, y_max = data[:, 1].min() - 1, data[:, 1].max() + 1
    return np.meshgrid(np.arange(x_min, x_max, 0.01),
                       np.arange(y_min, y_max, 0.01))

```

```

Отображаем данные
plt.scatter(train_data[:, 0], train_data[:, 1], c=train_labels, s=100,
           cmap='autumn')

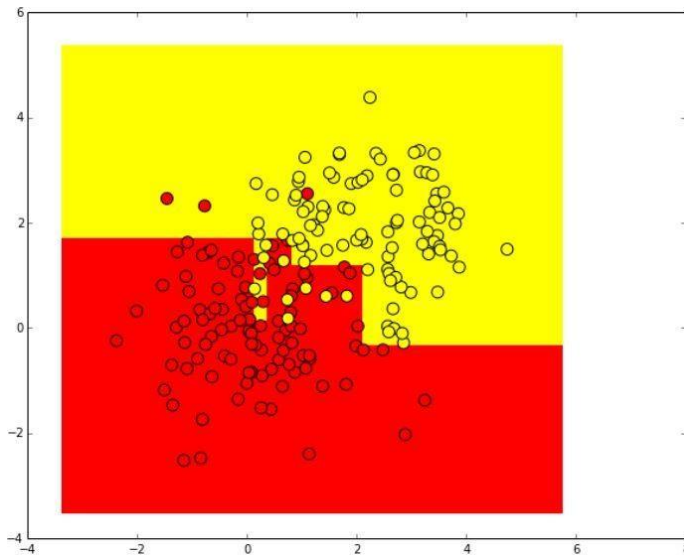
```



Обучение дерева решений. Разделение на классы. Визуализация границы разделения классов

```
from sklearn.tree import DecisionTreeClassifier
# параметр min_samples_leaf указывает, при каком минимальном количестве
# элементов в узле он будет дальше разделяться
clf = DecisionTreeClassifier(min_samples_leaf=5)
clf.fit(train_data, train_labels)

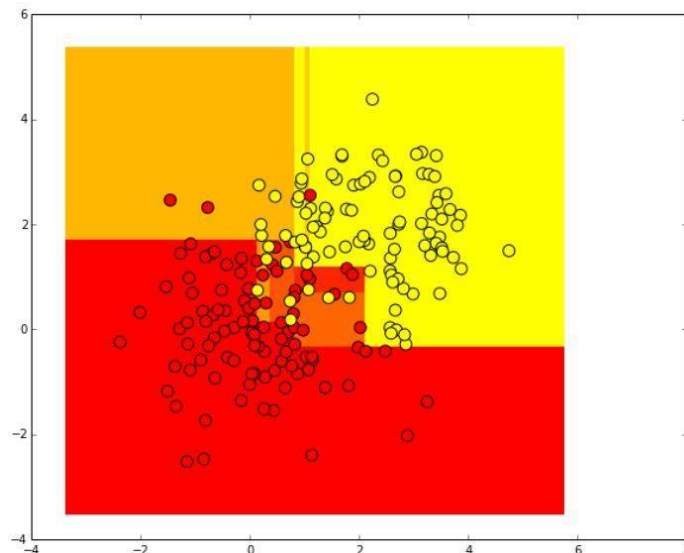
xx, yy = get_grid(train_data)
predicted = clf.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)
plt.pcolormesh(xx, yy, predicted, cmap='autumn')
plt.scatter(train_data[:, 0], train_data[:, 1], c=train_labels, s=100,
            cmap='autumn')
```



Предсказание вещественные вероятности (нормированные числа объектов разных классов из одного листа дерева) принадлежности первому классу.

Если в каком-то листе оказалось 5 положительных объектов (с меткой 1) и 2 отрицательных (с меткой 0), то вместо того, чтобы предсказывать метку 1 для всех объектов, которые попадут на этот лист, будет предсказываться величина $5/7$.

```
predicted_proba = clf.predict_proba(np.c_[xx.ravel(), yy.ravel()])[:,
1].reshape(xx.shape)
plt.pcolormesh(xx, yy, predicted_proba, cmap='autumn')
plt.scatter(train_data[:, 0], train_data[:, 1], c=train_labels, s=100,
cmap='autumn')
```

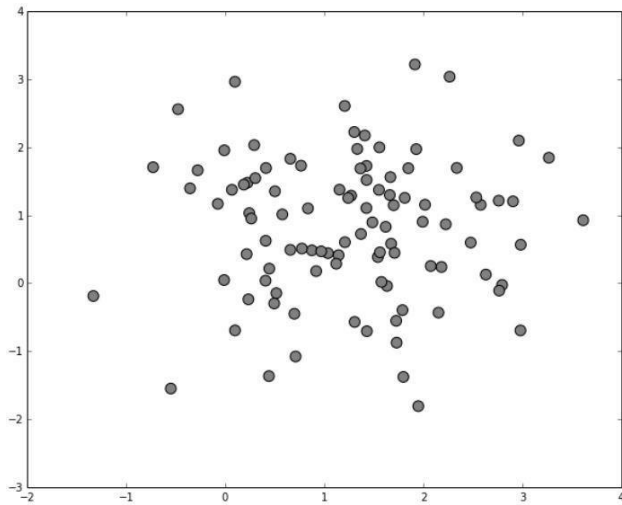


Сгенерируем случайно распределенные тестовые данные.

```
test_data = np.random.normal(size=(100, 2), loc = 1)
predicted = clf.predict(test_data)
```

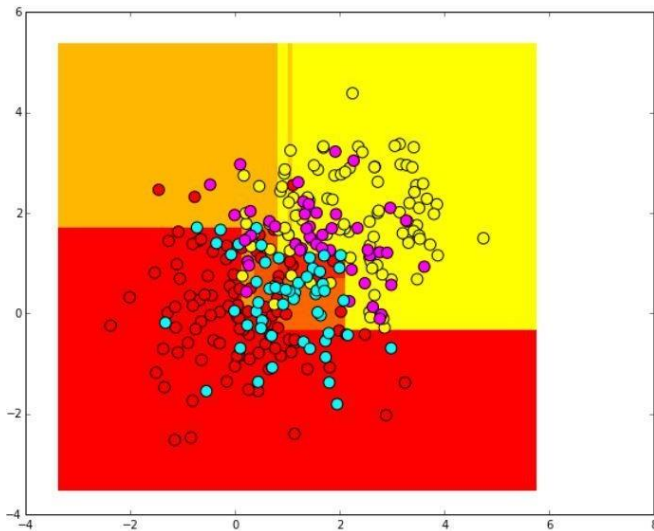
Отобразим их. Поскольку метки тестовых объектов неизвестны, нарисуем их серыми.

```
plt.scatter(test_data[:, 0], test_data[:, 1], c="gray", s=100)
```



Отобразим, как дерево решений классифицировало тестовые примеры. Для контраста необходимо выбрать другую цветовую гамму.

```
plt.pcolormesh(xx, yy, predicted_proba, cmap='autumn')
plt.scatter(train_data[:, 0], train_data[:, 1], c=train_labels, s=100,
            cmap='autumn')
plt.scatter(test_data[:, 0], test_data[:, 1], c=predicted, s=100,
            cmap='cool')
```



В итоге получили простой метод - дерево решений с использованием библиотеки Scikit-learn при решении задачи классификации.

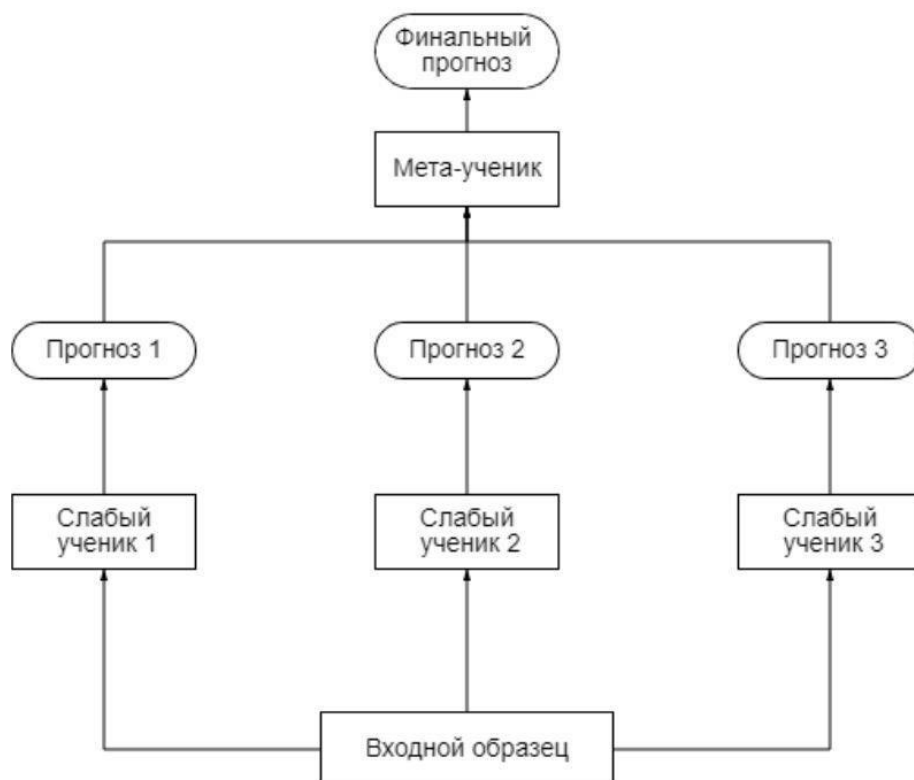
Семинарское занятие по теме 6

Тема занятия: Ансамблирование в машинном обучении

Задания (вопросы) для подготовки:

Стекинг

Приём удерживаемого набора на примере обучения мета-модели



На вход всех слабых прогнозаторов подаётся обучающий набор, каждый прогноз идёт к финальной модели, которая называется смеситель, мета-ученик или мета-модель, после чего та вырабатывает финальный прогноз.

Сначала набор разделяется на 2 части. Слабые ученики обучаются на первой половине обучающего набора, затем на второй. Затем создаётся новый обучающий набор на основе прогнозов, сделанных на прогнозах первой и второй части набора. Таким образом, на каждый образец из входного набора приходится столько прогнозов, сколько слабых учеников в ансамбле (в примере на картинке три). Мета-модель учится прогнозировать значения на основе нового набора.

```
from sklearn.ensemble import StackingClassifier
from sklearn.datasets import load_breast_cancer
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split

data, target = load_breast_cancer(return_X_y=True)

estimators = [('lr', LogisticRegression()), ('dt', DecisionTreeClassifier())]
modelClf = StackingClassifier(estimators=estimators, final_estimator=SVC())

X_train, X_valid, y_train, y_valid = train_test_split(data, target,
test_size=0.3, random_state=12)
```

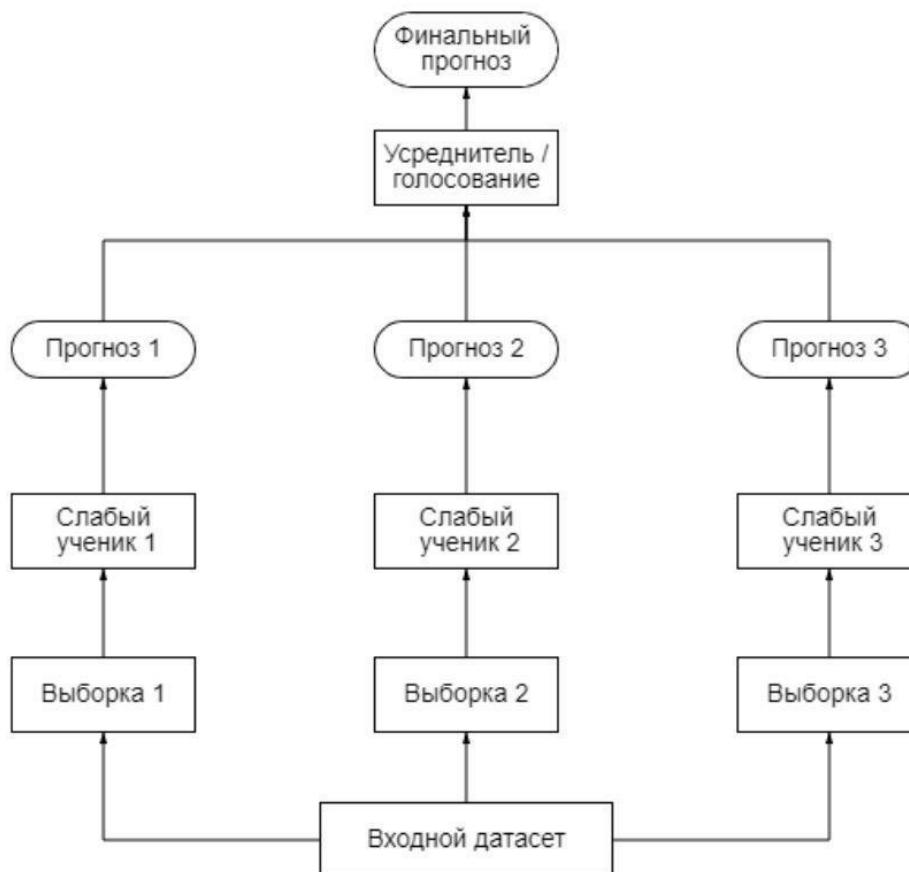
```
modelClf.fit(X_train, y_train)
print(modelClf.score(X_valid, y_valid))
```

Вывод:

0.9415204678362573

Бэггинг

Обучение нескольких одинаковых моделей на разных образцах. Распределение выборки неизвестно, поэтому модели получатся разными. Для начала генерируется несколько бутстрэп-выборок. Бутстрэп - это случайный выбор данных из датасета и представление их в модель, затем данные возвращаются в датасет и процесс повторяется. После модели делают свои прогнозы на основе бутстрэп-выборок. В случае регрессии прогнозы просто усредняются. В случае же классификации применяется голосование.



```

from sklearn.datasets import
load_breast_cancer from
sklearn.linear_model import
LogisticRegression from
sklearn.model_selection import
train_test_split from sklearn.ensemble
import BaggingClassifier

data, target = load_breast_cancer(return_X_y=True)

modelClf =
BaggingClassifier(base_estimator=LogisticRegression(),
n_estimators=50, random_state=12)

X_train, X_valid, y_train, y_valid =
train_test_split(data, target, test_size=0.3,
random_state=12)

modelClf.fit(X_train, y_train)
print(modelClf.score(X_valid, y_valid))

```

Вывод:

0.9239766081871345

Бустинг

Метод бустинга: берётся множество одинаковых моделей и объединяется, чтобы получить сильного ученика. Модели приспособляются к данным последовательно, то есть каждая модель исправляет ошибки предыдущей. Базовые модели для бустинга - это модели с низким разбросом и высоким смещением. Бустинг нельзя распараллелить.

Адаптивный бустинг (AdaBoost)

Данный алгоритм сначала обучает первую базовую модель (допустим деревья решений) на тренировочном наборе. Относительный вес некорректно предсказанных значений увеличивается. На вход второй базовой модели подаются обновлённые веса и модель обучается, после чего вырабатываются прогнозы и цикл повторяется. Результат работы AdaBoost - это средневзвешенная сумма каждой модели. Спрогнозированным значением ансамбля будет тот, который получает большинство взвешенный голосов:

$$C = \sum_{i=1}^N W * X$$

C - результат работы ансамбля, W - вес,

X - значение прогнозатора

Adaboost обновляет веса объектов на каждой итерации. Веса хорошо классифицированных объектов уменьшаются относительно весов неправильно

классифицированных объектов. Модели, которые работают лучше, имеют больший вес в окончательной модели ансамбля. При адаптивном бустинге используется *итеративный метод* (добавляем слабых учеников одного за другим, просматривая каждую итерацию, чтобы найти наилучшую возможную пару (коэффициент, слабый ученик) для добавления к текущей модели ансамбля) изменения весов. Он работает быстрее, чем аналитический метод.

```
from sklearn.datasets import
load_breast_cancer from sklearn.tree
import DecisionTreeClassifier from
sklearn.model_selection import
train_test_split from sklearn.ensemble
import AdaBoostClassifier

data, target =

load_breast_cancer(return_X_y=True)

modelClf =
AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=2),
n_estimators=100, random_state=12)

X_train, X_valid, y_train, y_valid =
train_test_split(data, target, test_size=0.3,
random_state=12)

modelClf.fit(X_train, y_train)
print(modelClf.score(X_valid, y_valid))
```

Вывод:
0.9590643274853801

Градиентный бустинг

Градиентный бустинг обучает слабые модели последовательно, исправляя ошибки предыдущих. Результатом градиентного бустинга также является средневзвешенная сумма результатов моделей. Принципиальное отличие от Adaboost это способ изменения весов. Адаптивный бустинг использует *итеративный метод* оптимизации. Градиентный бустинг оптимизируется с помощью градиентного спуска. Таким образом градиентный бустинг - обобщение адаптивного бустинга для дифференцируемых функций.

```
from sklearn.datasets import
load_breast_cancer from
sklearn.model_selection import
train_test_split from sklearn.ensemble import

GradientBoostingClassifier data, target =

load_breast_cancer(return_X_y=True)

modelClf = GradientBoostingClassifier(max_depth=2,
n_estimators=150, random_state=12, learning_rate=1)
```

```
X_train, X_valid, y_train, y_valid =
train_test_split(data, target, test_size=0.3,
random_state=12)
```

```
modelClf.fit(X_train, y_train)
print(modelClf.score(X_valid, y_valid))
```

Вывод:

0.935672514619883

Семинарское занятие по теме 8

Тема занятия: Ансамблирование в машинном обучении

Задания (вопросы) для подготовки:

Линейная модель

Полиномиальная модель (Poly2)

Факторизационная машина

Факторизационная машина с полями

5.2. Задания и упражнения для самостоятельной работы

| № п/п | Наименование темы дисциплины | Задания и упражнения для самостоятельной работы |
|-------|------------------------------|---|
| 1 | | |
| 2 | | |
| 3 | | |

5.4. Перечень тем (задания) для курсовой работы

Не предусмотрено

6.Оценочные средства для проведения текущего контроля и промежуточной аттестации обучающихся

6.1 Примерные оценочные средства, включая тестовые оценочные задания для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю) приведены в Приложении 1 к рабочей программе дисциплины.

6.2 Оценочные средства для проведения промежуточной аттестации обучающихся по дисциплине (модулю) включают следующие разделы:

- перечень компетенций, формируемых в процессе освоения учебной дисциплины;
- описание показателей и критериев оценивания компетенций, описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения по учебной дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы;
- методические материалы, определяющие процедуры оценивания результатов обучения по учебной дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

7.Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Тимофеев, А. В. Проектирование и разработка информационных систем : учебное пособие для СПО / А. В. Тимофеев, З. Ф. Камальдинова, Н. С. Агафонова. — Саратов : Профобразование, 2022. — 91 с. — ISBN 978-5-4488-1416-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL:

<https://www.iprbookshop.ru/116285.html> (дата обращения: 03.02.2022). — Режим доступа: для авторизир. пользователей

2. Ахмадиев, Ф. Г. Математическое моделирование и методы оптимизации : учебное пособие / Ф. Г. Ахмадиев, Р. М. Гильфанов. — Москва : Ай Пи Ар Медиа, 2022. — 178 с. — ISBN 978-5-4497-1383-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/116448.html> (дата обращения: 03.02.2022). — Режим доступа: для авторизир. пользователей

б) дополнительная литература:

1. Орлов, А. И. Основы теории принятия решений : учебное пособие / А. И. Орлов. — Москва : Ай Пи Ар Медиа, 2022. — 66 с. — ISBN 978-5-4497-1423-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/117037.html> (дата обращения: 03.02.2022). — Режим доступа: для авторизир. пользователей

в) интернет-ресурсы:

1. ЭБС IPR Books <http://iprbookshop.ru> (учебники и учебные пособия, монографии, сборники научных трудов, научная периодика, профильные журналы, справочники, энциклопедии);
2. ООО «ИВИС» <https://dlib.eastview.com> (электронные версии периодических изданий ООО «ИВИС»);
3. Web of Science <http://webofknowledge.com/> (обширная международная универсальная реферативная база данных);
4. Scopus <https://www.scopus.com> (международная универсальная реферативная база данных, индексирующая более 21 тыс. наименований научно-технических, гуманитарных и медицинских журналов, материалов конференций примерно 5000 международных издательств);
5. Научная электронная библиотека eLIBRARY.RU <https://elibrary.ru> (крупнейший российский информационный портал в области науки, технологии, медицины и образования);
6. Единая межведомственная информационно-статистическая система (ЕМИСС) <https://fedstat.ru/indicators/>
7. Федеральная служба государственной статистики <http://www.gks.ru/>
8. Министерство науки и высшего образования Российской Федерации <https://minobrnauki.gov.ru/>
9. Министерство экономического развития Российской Федерации <http://economy.gov.ru/minec/about/structure>
10. Федеральное агентство по печати и массовым коммуникациям <http://www.farpmc.ru/rospechat.html>
11. Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации <https://digital.gov.ru/ru>
12. <http://www.mathnet.ru/> Общероссийский математический портал (информационная система)

8. Материально-техническое обеспечение дисциплины

| № | Наименование учебных аудиторий (лабораторий) и помещений для самостоятельной работы | Перечень оборудования и технических средств обучения | Программное обеспечение, в т.ч. отечественного производства |
|---|---|---|---|
| 1 | Учебные аудитории для проведения занятий лекционного типа | Преподавательский стол; столы обучающихся; стулья; классная доска; мультимедийный комплекс; | 1. 7-Zip (Бесплатное ПО); 2. 10-Strike Network Inventory ПО РФ (ПО) 3. Ductor Academic ПО РФ (Бесплатное ПО); https://basegroup.ru/deductor/manual/licence-deductor-academic 4. Eset Endpoint security (Платное ПО) https://help.eset.com/eula/ GIMP (Бесплатное ПО); https://docs.gimp.org/2.10/ru/ |

| | | | |
|---|--------------------|---|---|
| | | <p>наглядные пособия (плакаты) Место, оборудованное для лиц с ограниченными возможностями и.</p> <p>Лицензионное программное обеспечение, подключенное к Интернет</p> | <p>5. microsoft office проффессиональный плюс 2016 (ПО) https://www.microsoft.com/en-us/Useterms/Retail/Office/2016Professional/Useterms_Retail_Office_2016Professional_RUS.htm</p> <p>6. Microsoft power Bi (Бесплатное ПО); https://powerbi.microsoft.com/ru-ru/windows-license-terms/ icrosoft Visual Studio (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mt171584/</p> <p>7. Notepad ++ (Бесплатное ПО); https://www.gnu.org/licenses/old-licenses/gpl-2.0.html</p> <p>8. Zoom (Бесплатное ПО); https://explore.zoom.us/ru/terms/</p> <p>9. Anaconda3 2019 (Бесплатное ПО);</p> <p>10. https://www.anaconda.com/eula-anaconda-individual-edition Android studio (Бесплатное ПО); https://developer.android.com/studio/terms</p> <p>11. Brackets (Бесплатное ПО); https://github.com/brackets-cont/brackets/blob/master/LICENSE</p> <p>12. CodeBlocks (Бесплатное ПО);https://www.codeblocks.org/license/ Firebird (Бесплатное ПО); https://firebirdsql.org/en/licensing/</p> <p>13. KNIME analytics platform (Бесплатное ПО); https://www.knime.com/downloads/full-license</p> <p>14. Loginom community РФ ПО (Бесплатное ПО);https://loginom.ru/legal</p> <p>15. Monogame SDK (Бесплатное ПО); https://github.com/MonoGame/MonoGame/blob/develop/LICENSE.txt Openproj (Бесплатное ПО); https://opensource.org/licenses/CPAL-1.0</p> <p>16. tableau 2019 (Бесплатное ПО); https://www.tableau.com/legal</p> <p>17. Visual studio community 2017 (Бесплатное ПО); https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi-qZeRxv7zAhXhsYsKHZoRBAsQFnoECBgQAQ&url=https%3A%2F%2Fvisualstudio.microsoft.com%2Fwp-content%2Fuploads%2F2017%2F01%2FVS2017_COMMUNITY_RC_RUS_Eula.1049-1.docx&usg=AOvVaw0tLx1QA4E2McNypfRn9tTo</p> <p>18. Visual studio community 2019 (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mlt110718/</p> <p>19. Консультант плюс</p> |
| 2 | Компьютерный класс | <p>Преподавательский стол; столы обучающихся; стулья; классная доска; мультимедийный комплекс; ПК преподавателя; ПК обучающихся; наглядные пособия (плакаты) Место, оборудованное для лиц с ограниченными возможностями и.</p> <p>Лицензионное программное обеспечение, подключенное к Интернет</p> | <p>1. 7-Zip (Бесплатное ПО);</p> <p>2. 10-Strike Network Inventory ПО РФ (ПО)</p> <p>3. Ductor Academic ПО РФ (Бесплатное ПО); https://basegroup.ru/deductor/manual/licence-deductor-academic</p> <p>4. Eset Endpoint security (Платное ПО) https://help.eset.com/eula/GIMP (Бесплатное ПО); https://docs.gimp.org/2.10/ru/</p> <p>5. microsoft office проффессиональный плюс 2016 (ПО) https://www.microsoft.com/en-us/Useterms/Retail/Office/2016Professional/Useterms_Retail_Office_2016Professional_RUS.htm</p> <p>6. Microsoft power Bi (Бесплатное ПО); https://powerbi.microsoft.com/ru-ru/windows-license-terms/ icrosoft Visual Studio (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mt171584/</p> <p>7. Notepad ++ (Бесплатное ПО); https://www.gnu.org/licenses/old-licenses/gpl-2.0.html</p> <p>8. Zoom (Бесплатное ПО); https://explore.zoom.us/ru/terms/</p> <p>9. Anaconda3 2019 (Бесплатное ПО);</p> <p>10. https://www.anaconda.com/eula-anaconda-individual-edition Android studio (Бесплатное ПО); https://developer.android.com/studio/terms</p> <p>11. Brackets (Бесплатное ПО); https://github.com/brackets-cont/brackets/blob/master/LICENSE</p> <p>12. CodeBlocks (Бесплатное ПО);https://www.codeblocks.org/license/ Firebird (Бесплатное ПО); https://firebirdsql.org/en/licensing/</p> |

| | | | |
|---|--|--|--|
| | | | <p>13. KNIME analytics platform (Бесплатное ПО); https://www.knime.com/downloads/full-license</p> <p>14. Loginom community РФ ПО (Бесплатное ПО); https://loginom.ru/legal</p> <p>15. Monogame SDK (Бесплатное ПО); https://github.com/MonoGame/MonoGame/blob/develop/LICENSE.txt</p> <p>Openproj (Бесплатное ПО); https://opensource.org/licenses/CPAL-1.0</p> <p>16. tableau 2019 (Бесплатное ПО); https://www.tableau.com/legal</p> <p>17. Visual studio community 2017 (Бесплатное ПО); https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi-qZeRxv7zAhXhsYsKHZoRBAsQFnoECBgQAQ&url=https%3A%2F%2Fvisualstudio.microsoft.com%2Fwp-content%2Fuploads%2F2017%2F01%2FVS2017_COMMUNITY_RC_RUS_Eula.1049-1.docx&usg=AOvVaw0tLx1QA4E2McNypfRn9tTo</p> <p>18. Visual studio community 2019 (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mlt110718/</p> <p>19. Консультант плюс</p> |
| 3 | <p>Учебные аудитории для проведения занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также самостоятельной работы обучающихся</p> | <p>Преподавательский стол; столы обучающихся; стулья; классная доска; мультимедийный комплекс; ПК преподавателя; ПК обучающихся; наглядные пособия (плакаты) <i>Место, оборудованное для лиц с ограниченными возможностями и.</i> Лицензионное программное обеспечение, подключенное к Интернет</p> | <p>1. 7-Zip (Бесплатное ПО);</p> <p>2. 10-Strike Network Inventory ПО РФ (ПО)</p> <p>3. Ductor Academic ПО РФ (Бесплатное ПО); https://basegroup.ru/deductor/manual/licence-deductor-academic</p> <p>4. Eset Endpoint security (Платное ПО) https://help.eset.com/eula/GIMP (Бесплатное ПО); https://docs.gimp.org/2.10/ru/</p> <p>5. microsoft office проффессиональный плюс 2016 (ПО) https://www.microsoft.com/en-us/Useterms/Retail/Office/2016Professional/Useterms_Retail_Office_2016Professional_RUS.htm</p> <p>6. Microsoft power Bi (Бесплатное ПО); https://powerbi.microsoft.com/ru-ru/windows-license-terms/ icrosoft Visual Studio (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mt171584/</p> <p>7. Notepad ++ (Бесплатное ПО); https://www.gnu.org/licenses/old-licenses/gpl-2.0.html</p> <p>8. Zoom (Бесплатное ПО); https://explore.zoom.us/ru/terms/</p> <p>9. Anaconda3 2019 (Бесплатное ПО);</p> <p>10. https://www.anaconda.com/eula-anaconda-individual-edition Android studio (Бесплатное ПО); https://developer.android.com/studio/terms</p> <p>11. Brackets (Бесплатное ПО); https://github.com/brackets-cont/brackets/blob/master/LICENSE</p> <p>12. CodeBlocks (Бесплатное ПО);https://www.codeblocks.org/license/ Firebird (Бесплатное ПО); https://firebirdsql.org/en/licensing/</p> <p>13. KNIME analytics platform (Бесплатное ПО); https://www.knime.com/downloads/full-license</p> <p>14. Loginom community РФ ПО (Бесплатное ПО); https://loginom.ru/legal</p> <p>15. Monogame SDK (Бесплатное ПО); https://github.com/MonoGame/MonoGame/blob/develop/LICENSE.txt</p> <p>Openproj (Бесплатное ПО); https://opensource.org/licenses/CPAL-1.0</p> <p>16. tableau 2019 (Бесплатное ПО); https://www.tableau.com/legal</p> <p>17. Visual studio community 2017 (Бесплатное ПО); https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi-qZeRxv7zAhXhsYsKHZoRBAsQFnoECBgQAQ&url=https%3A%2F%2Fvisualstudio.microsoft.com%2Fwp-content%2Fuploads%2F2017%2F01%2FVS2017_COMMUNITY_RC_RUS_Eula.1049-1.docx&usg=AOvVaw0tLx1QA4E2McNypfRn9tTo</p> <p>18. Visual studio community 2019 (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mlt110718/</p> <p>19. Консультант плюс</p> |

| | | | |
|---|------------------------------|--|--|
| 4 | Библиотека с читальным залом | Стол обучающегося, стулья, ПК обучающегося, принтер Электронная библиотечная система и библиотечное абонентное обслуживание (учебная литература на бумажных носителях) Лицензионное программное обеспечение, подключенное к Интернет | <ol style="list-style-type: none"> 1. 7-Zip (Бесплатное ПО); 2. microsoft office проффессиональный плюс 2016 (ПО) https://www.microsoft.com/en-us/Useterms/Retail/Office/2016Professional/Useterms_Retail_Office_2016Professional_RUS.htm 3. Microsoft power Bi (Бесплатное ПО); https://powerbi.microsoft.com/ru-ru/windows-license-terms/ icrosoft Visual Studio (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mt171584/ 4. Антиплагиат 5. Консультант плюс |
|---|------------------------------|--|--|

9.Перечень информационных технологий

Образовательный процесс по дисциплине поддерживается средствами электронной информационно-образовательной среды Университета, которая обеспечивает:

- доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем и электронным образовательным ресурсам, указанным в рабочей программе, через личный кабинет студента и преподавателя;
- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения основной образовательной программы;
- проведение всех видов занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением дистанционных образовательных технологий;
- формирование электронного портфолио обучающегося, в том числе сохранение работ обучающегося, рецензий и оценок на эти работы со стороны любых участников образовательного процесса;
- взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействие посредством сети Интернет.

Каждый обучающийся обеспечен индивидуальным неограниченным доступом к электронно-библиотечной системе (ЭБС университета), содержащей издания учебной, учебно-методической и иной литературы по основным изучаемым дисциплинам и сформированной на основании прямых договоров с правообладателями.

Перечень программного обеспечения:

1. Adobe flash player 31;
2. Adobe reader 10;
3. Java 6.0;
4. K-Lite Codec Pack;
5. Win rar;
6. Microsoft Office 10;
7. Microsoft Visio 10;
8. Microsoft Visual studio;

Профессиональные базы данных и информационно-справочные системы:

1. Kaspersky Endpoint Security для бизнеса <http://inion.ru/resources/bazy-dannykh-inion-ran/> - библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам;
2. <http://inion.ru/resources/bazy-dannykh-inion-ran/> - библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам;
3. <http://www.scopus.com/> - реферативная база данных Scopus – международная

универсальная реферативная база данных;

4. <http://elibrary.ru/defaultx.asp> - крупнейший российский информационный портал электронных журналов и баз данных по всем отраслям наук.

5. <https://www.cfin.ru/finanalysis/math/>, База данных «Финансовая математика – Библиотека управления» - Корпоративный менеджмент

10. Методические указания для обучающихся

10.1. Преподавание дисциплины осуществляется в соответствии с Федеральным государственным образовательным стандартом высшего образования

Основными формами получения и закрепления знаний по данной дисциплине являются занятия лекционного и семинарского типа, самостоятельная работа обучающегося, в том числе под руководством преподавателя, прохождение рубежного контроля.

Основной объем часов по изучению дисциплины согласно учебным планам приходится на самостоятельную работу обучающихся. Самостоятельная работа включает в себя изучение учебной, учебно-методической и специальной литературы, её конспектирование, подготовку к занятиям семинарского типа, текущему контролю и промежуточной аттестации (зачету или (и) экзамену).

Текущий контроль успеваемости по учебной дисциплине и промежуточная аттестация осуществляются в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования: программам бакалавриата, программам специалитета, программам магистратуры.

Наличие в Университете электронной информационно-образовательной среды, а также электронных образовательных ресурсов позволяет осваивать курс инвалидам и лицам с ОВЗ.

10.2. Особенности освоения учебной дисциплины инвалидами и лицами с ограниченными возможностями здоровья.

Особенности освоения учебной дисциплины инвалидами и лицами с ОВЗ определены в Положении об организации обучения студентов-инвалидов и студентов с ограниченными возможностями здоровья, утвержденным приказом ректора.

Обучение инвалидов и лиц с ОВЗ может осуществляться индивидуально, а также с применением электронного обучения, дистанционных образовательных технологий.

Выбор методов и средств обучения, образовательных технологий и учебно-методического обеспечения реализации образовательной программы осуществляется Университетом самостоятельно, исходя из необходимости достижения обучающимися планируемых результатов освоения образовательной программы, а также с учетом индивидуальных возможностей обучающихся из числа инвалидов и лиц с ОВЗ.

Форма проведения промежуточной аттестации для студентов-инвалидов и лиц с ОВЗ устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При необходимости инвалидам и лицам с ОВЗ предоставляется дополнительное время для подготовки ответа на зачете или экзамене.

В группах, в состав которых входят студенты с ОВЗ, с целью реализации индивидуального подхода, а также принципа индивидуализации и дифференциации, рекомендуется использовать технологию нелинейной конструкции учебных занятий, предусматривающую одновременное сочетание фронтальных, групповых и индивидуальных форм работы с различными категориями студентов, в т.ч. имеющих ОВЗ. В случае наличия обучающихся с нарушением функций опорно-двигательного аппарата, зрения и слуха, они обеспечиваются необходимым оборудованием, имеющимся в Университете, а также предоставляемым в рамках Соглашения с РУМЦ РГСУ от 14 ноября

2019 года.

11. Методические рекомендации преподавателю по организации учебного процесса по дисциплине

При изучении дисциплины рекомендуется использовать следующий набор средств и способов обучения:

- рекомендуемую основную и дополнительную литературу;
 - задания для подготовки к занятиям семинарского типа (вопросы для обсуждения, кейс задания, расчетные задачи и др.);
 - задания для текущего контроля успеваемости (задания для самостоятельной работы обучающихся, тестовые задания в рамках электронной системы тестирования);
 - вопросы и задания для подготовки к промежуточной аттестации по итогам освоения дисциплины, позволяющие оценить знания, умения и уровень приобретенных компетенций.
- При проведении занятий лекционного и семинарского типа, в том числе в форме вебинаров и on-line курсов необходимо строго придерживаться тематического плана дисциплины, приведенного в РПД. Необходимо уделить внимание рассмотрению вопросов и заданий, включенных в тестовые оценочные задания, при необходимости, решить аналогичные задачи с объяснением алгоритма решения.

Следует обратить внимание обучающихся на то, что для успешной подготовки к текущему контролю (выполнению ТОЗ) и промежуточной аттестации (зачету или экзамену) недостаточно прочесть рабочий учебник, размещенный в личном кабинете. Нужно изучить материалы основной и дополнительной литературы, список которой приведен в РПД, законодательные и нормативные акты, а также материалы, рекомендованные в разделе «Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины».

Текущий контроль успеваемости по учебной дисциплине и промежуточная аттестация осуществляются в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования: программам бакалавриата, программам специалитета, программам магистратуры.

Программа разработана Замега Э.Н.

Рабочая программа дисциплины рассмотрена и принята на заседании кафедры от 27.01.2022 г., протокол №6.

Лист регистрации изменений и дополнений в рабочую учебную программу

Составителем внесены следующие изменения:

| Содержание изменений | Номер протокола и дата заседания кафедры, по утверждению изменений |
|--|---|
| Рабочая программа дополнена и переутверждена | Протокол №1 от 28.08.2023г. |
| Рабочая программа дополнена и переутверждена | Протокол №1 от 29.08.2024г. |