

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор по цифровизации
образования**

Д.И. Гриц

| | |
|----------------------------|---|
| | Рабочая программа дисциплины (модуля) |
| по дисциплине: | Основы машинного и глубокого обучения |
| по направлению: | Прикладные математика и физика |
| профиль подготовки: | Природоподобные технологии и биомиметический дизайн материалов и систем Физтех-школа природоподобных, плазменных и ядерных технологий им. И.В. Курчатова центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" |
| курс: | 1 |
| квалификация: | магистр |

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Зачет
- 2 (весенний) - Зачет

Аудиторных часов: 90 всего, в том числе:

- лекции: 30 час.
- семинары: 60 час.
- лабораторные занятия: 0 час.

Самостоятельная работа: 180 час.

Всего часов: 270, всего зач. ед.: 6

Количество контрольных работ, заданий: 2

Программу составили:

Р.Г. Нейчев, старший преподаватель
А.П. Чистов, ассистент
М.А. Певцова, методист
Ж.И. Зубцова, канд. физ.-мат. наук, эксперт

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и онлайн-образования "Пуск" 28.01.2025

Аннотация

Дисциплина состоит из двух модулей:

Модуль 1. Основы программирования на Python для машинного обучения

Модуль 2. Основы машинного обучения

По итогам обучения обучающийся будет способен формализовать и алгоритмизировать поставленную задачу, написать программный код с использованием языков программирования, оформить код в соответствии с установленными требованиями.

1. Цели и задачи

Цель дисциплины

- совершенствование компетенций студентов в области решения профессиональных задач по работе с данными с помощью основных методов машинного и глубокого обучения.

Задачи дисциплины

- научиться использовать базовые типы и конструкции языка программирования Python;
- сформировать умение работать со стандартными структурами данных в Python, писать функции на Python, применять функциональные особенности языка, работать с файлами с помощью языка Python;
- научиться применять механизмы наследования, создавать классы и работать с ними, обрабатывать исключения;
- сформировать умение искать и исправлять ошибки в программе на Python, тестировать программы на Python;
- научиться писать многопоточный код на Python, писать асинхронный код на Python, работать с сетью, создать свое серверное сетевое приложение;
- сформировать умение пользоваться библиотеками Python для работы с данными;
- научиться решать оптимизационные задачи с помощью Python;
- сформировать умение использовать математический аппарат для работы с данными;
- приобрести навыки построения предсказывающих моделей;
- сформировать умение оценивать качество построенных моделей;
- научиться применять инструменты Python для решения задач машинного обучения.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

| Код и наименование компетенции | Индикаторы достижения компетенции |
|--|---|
| УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной задачи | УК-3.1 Организует и координирует работу участников проекта, способствует конструктивному преодолению возникающих разногласий и конфликтов |
| | УК-3.4 Способен планировать командную работу, распределять поручения членам команды, организовать обсуждение разных идей и мнений |
| ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий | ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов |

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- принципы организации кода;
- базовые типы и конструкции языка Python;
- понятия вектора и матрицы, векторного пространства, нормы вектора, ортогональности и гиперплоскости;
- как выполнять базовые операции над векторами и матрицами;
- как библиотека NumPy помогает в научных вычислениях и обработке данных;
- назначение и принцип работы библиотеки NumPy;
- основные методы обработки данных, упрощающие решение задач машинного обучения;
- в каких сферах применяется машинное обучение;
- основные понятия машинного обучения: датасет (выборка), объект, признак, таргет, матрица объект-признак, машинное обучение с учителем, таргет, модель, предсказание, функция потерь, параметр, гиперпараметр;
- формальную постановку задачи машинного обучения с учителем;
- как строится сверточная нейронная сеть;
- часто используемые в CNN техники (padding, striding, pooling);
- различные архитектуры, используемые в машинном обучении;
- понятие коллекции;
- назначение коллекций в разработке;
- в каких задачах машинного обучения используются линейные модели;
- теорему Гаусса-Маркова;
- основные понятия теории вероятностей;
- понятие условной вероятности, дискретных и непрерывных случайных величин;
- центральную предельную теорему и теорему Байеса;
- в каких задачах можно применить наивный Байесовский классификатор;
- как строится рекуррентная нейронная сеть (RNN);
- разновидностями архитектуры RNN — LSTM и GRU;
- проблемы затухающих и взрывающихся градиентов;
- понятие и роль специальных методов классов в программировании на Python;
- способы подключения модулей стандартной библиотеки;
- модули стандартной библиотеки;
- виды графического представления данных и ситуации их использования;
- возможности библиотеки matplotlib для построения различных видов графиков и настройки их отображения;
- функции и их свойства;
- как находить производные функции, что такое экстремумы и критические точки функции;
- что такое градиент функции;
- методы оптимизации;
- задачу обработки естественного языка;
- понятия правдоподобия в задачах машинного обучения;
- различные метрики оценки качества классификации;
- основные подходы градиентной оптимизации;
- основные техники регуляризации в глубоком обучении;
- метод опорных векторов, используемый для задач классификации и регрессионного анализа;
- способы создания нелинейного классификатора с помощью так называемого ядерного трюка (kernel trick);
- критерии информативности: энтропия и критерий Джини;
- как использовать решающие деревья в задаче регрессии;
- основные нелинейные функции активации;
- механизмы обратного распространения ошибки;
- принципы работы с функциями: синтаксис функций, способы задания аргументов и возврата значений;
- понятия классов и объектов в Python, их взаимосвязь;
- особенности объектно-ориентированной модели в Python;
- понятие и механизмы наследования, его роль в программировании;
- подходы к обработке ошибок;
- механизм формирования исключений;
- возможности библиотеки seaborn для построения различных видов графиков;
- основы математической статистики;
- различные техники ансамблирования и теоретические предпосылки к их применению;
- процедуру bootstrap, на основе которой строится метод ансамблирования бэггинг;
- принципы построения случайного леса;
- как использовать технику блендинга, которая позволяет строить ансамбли не параллельно друг другу, а последовательно;

уметь:

- Выполнять практические задачи и проекты в команде;
- Установить интерпретатор Python на компьютер;
- Установить среду разработки PyCharm;
- Написать простой код на Python;
- использовать базовые типы и конструкции Python для написания простых программ;
- применять NumPy для работы с векторами и матрицами;
- применять метод kNN для решения задач машинного обучения;
- обучать сверточную нейронную сеть (CNN);
- использовать сверточную нейронную сеть для обработки изображений;
- использовать встроенные в Python коллекции для написания программ;
- сопоставлять и выбрать необходимую структуру данных для конкретной практической задачи;
- формально поставить задачу линейной регрессии;
- использовать L1- и L2-регуляризации для решения задач машинного обучения;
- использовать специальные методы в написании программ;
- применять функции для построения основных видов графиков и настраивать внешний вид графиков (цвет, подписи, легенда, сетка);
- строить 3D-изображения с помощью библиотек Python и использовать их для задач компьютерного зрения;
- решать задачи оптимизации градиентными методами;
- предобрабатывать текстовые данные и извлекать из них признаки;
- строить информативные векторные представления слов;
- решать задачи линейной классификации в машинном обучении;
- использовать модель логистической регрессии в задачах бинарной и мультиклассовой классификации;
- бороться с переобучением нейросети с помощью регуляризации;
- строить модель с регуляризацией в PyTorch;
- использовать решающие деревья в задачах машинного обучения;
- описывать линейные и нелинейные зависимости с помощью нейросетей;
- строить простейшую нейросеть на PyTorch;
- использовать функции как объект, функции высших порядков, лямбда-функции;
- выполнить чтение и запись в файл;
- применять рекомендации PEP 8 для написания кода;
- создавать классы и использовать методы классов;
- проектировать необходимые классы и методы классов;
- проектировать классы с использованием механизма наследования;
- использовать конструкции языка для генерации исключений на Python;
- создавать виртуальное окружение;
- выполнять статистический анализ программного кода;
- работать с распределенной системой контроля версий Git;
- получить информацию о DataFrame, вычислить описательные статистики для числовых данных, обратиться к элементам DataFrame по индексу и порядковому номеру, изменить индекс;
- выполнять поиск, фильтрацию и сортировку DataFrame с применением методов библиотеки Pandas;
- вычислять статистику по признакам, применять функции к данным, рассчитывать новые значения;
- работать с несколькими таблицами с помощью инструментов библиотеки Pandas;
- строить графики в Pandas;
- подбирать методы, признаки и число кластеров для кластеризации;
- оценивать влияние признаков на прогноз модели машинного обучения с помощью библиотеки SHAP;
- использовать метод кросс-валидации для оценки качества модели;
- пользоваться EM-алгоритмом;
- пользоваться методом DBSCAN;
- решать задачи классификации и регрессии с помощью градиентного бустинга;
- использовать градиентный бустинг;
- эффективно использовать большие объемы данных посредством понижения их размерности.

владеть:

- стандартными структурами данных в Python, умением писать функции на Python, применять функциональные особенности языка, работать с файлами с помощью языка Python;
- механизмами наследования, создавать классы и работать с ними, обрабатывать исключения;
- навыками выбора подходящего метода оптимизации для конкретной задачи;
- навыками применения библиотеки Python для построения модели линейной регрессии, решающих деревьев и композиций алгоритмов, для обучения метрических алгоритмов, SVM, байесовских моделей.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

| № | Тема (раздел) дисциплины | Трудоемкость по видам учебных занятий, включая самостоятельную работу, час. | | | |
|-----------------------|--|---|----------|-----------------|----------------|
| | | Лекции | Семинары | Лаборат. работы | Самост. работа |
| 1 | Основы программирования на Python для машинного обучения | 15 | 30 | | 90 |
| 2 | Основы машинного обучения | 15 | 30 | | 90 |
| Итого часов | | 30 | 60 | | 180 |
| Подготовка к экзамену | | 0 час. | | | |
| Общая трудоёмкость | | 270 час., 6 зач.ед. | | | |

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

1. Основы программирования на Python для машинного обучения

1.1. Введение в программирование на Python

Лекция

О языке Python

Работа в IDE PyCharm. Первая программа

Введение в Python. Ввод и вывод данных

Типы данных и операции над ними

Условный оператор

Цикл while

Цикл for

Операторы continue и break

Практическая работа

Семинар "Введение в программирование на Python"

Самостоятельная работа

Дополнительные материалы

Установка интерпретатора Python

Тест для самопроверки

Задания на программирование

1.2. Структуры данных и функции

1.2.1. Коллекции

Лекция

Встроенные структуры данных

Множества и примеры работы с множествами
Строки и примеры работы со строками
Списки и примеры работы со списками
Методы split() и join() и примеры работы с ними
Кортежи
Словари
Практическая работа
Семинар "Коллекции"
Самостоятельная работа
Дополнительные материалы
Методы списков и строк
Тест для самопроверки
Задания на программирование
1.2.2. Функции. Работа с файлами
Лекция
Именные функции, инструкция def
Возврат значений из функции
Множественное присваивание, упаковка и распаковка значений
Аргументы по умолчанию и именованные аргументы
Инструкция pass(). Согласованность аргументов
Функция как объект. Функции высших порядков
Лямбда-функция
Принципы работы с файлами на Python
Разбор задач на работу с файлами
Правила записи кода PEP 8
Практическая работа
Семинар "Функции"
Самостоятельная работа
Дополнительные материалы
Тест для самопроверки
Задания на программирование
1.3. Объектно-ориентированное программирование
Лекция
Введение в объектно-ориентированное программирование
Классы и экземпляры классов
Методы. Пример рефакторинга программы на ООП
Инкапсуляция. Полиморфизм
Наследование классов
Особенности объектной модели в Python
Элементы статической типизации. Абстрактные классы и протоколы
Множественное наследование
Проблемы, связанные с наследованием
Композиция классов
Практические рекомендации
Практическая работа
Семинар "Классы и объекты"
Самостоятельная работа
Дополнительные материалы
Тест для самопроверки
Задания на программирование
1.4. Углубленный Python
1.4.1. Работа с ошибками
Лекция
Обработка ошибок в программировании

Две основные стратегии обработки ошибок
Синтаксис обработки ошибок
Обработка исключений и производительность
Генерация исключений
Инструкция `assert`
Классы исключений
Создание пользовательских исключений
Практика работы с исключениями
Практическая работа
Семинар "Работа с ошибками"
Самостоятельная работа
Дополнительные материалы
Тест на самопроверку по уроку
Задания на программирование
1.4.2. Специальные методы классов и установка внешних библиотек
Лекция
Специальные методы классов
Хеширование
Специальные атрибуты
Перегрузка операторов
Коллекции и итераторы
Контекстные менеджеры
Инструкция `import`
Модули стандартной библиотеки
Создание своего модуля на Python
Установка внешних библиотек Python
Практическая работа
Семинар "Специальные методы классов"
Самостоятельная работа
Дополнительные материалы
Тест для самопроверки
Задания на программирование
1.5. Анализ данных с помощью прикладных библиотек Python
1.5.1. Основы линейной алгебры в NumPy
Лекция
Векторы. Основные операции над векторами
Матрицы. Основные операции над матрицами
Вычислительные функции библиотеки NumPy. Массивы
Векторы. Решение линейных уравнений в NumPy
Использование NumPy в задачах обработки данных. Генерация мелодии
Работа с табличными данными и векторами
Библиотека NumPy. Линейная алгебра в NumPy
Задача снижения размерности
Метод главных компонент
Практическая работа
Семинар "Основы линейной алгебры в NumPy"
Самостоятельная работа
Дополнительные материалы
Библиотека NumPy
Примеры работы с NumPy
Тест для самопроверки
Задания на программирование
1.5.2. Описательная статистика. Анализ данных с помощью Pandas
Лекция

Описательная статистика
Базовые операции с DataFrame
Работа с пропусками и операции над данными
Работа с несколькими таблицами (Join)
Построение графиков в Pandas
Основы математической статистики
Практическая работа
Семинар "Описательная статистика. Анализ данных с помощью Pandas"
Самостоятельная работа
Дополнительные материалы
Построение графиков в Pandas
Анализ данных в Pandas
Тест для самопроверки
Задания на программирование
1.5.3. Построение графиков с помощью Matplotlib
Лекция
Методы визуализации
Библиотека Matplotlib
Основы построения графиков с помощью Matplotlib
Анатомия графиков в Matplotlib
Несколько областей рисования с помощью Matplotlib
3D-визуализация графики для машинного обучения
Работа с изображением в NumPy
Практическая работа
Семинар "Построение графиков с помощью Matplotlib. Методы регрессионного анализа"
Самостоятельная работа
Дополнительные материалы
Тест для самопроверки
Задания на программирование
Тест на оценку по теме "Анализ данных с помощью прикладных библиотек Python"
1.5.4. Методы регрессионного анализа
Лекция
Линейные модели машинного обучения
Линейная регрессия
Теорема Гаусса-Маркова
L1 и L2 регуляризация
Решение линейной регрессии и анализ устойчивости решения
Интерполяция данных
Простая линейная регрессия
Линейная регрессия с помощью sklearn
Практическая работа
Семинар "Генетические алгоритмы и оптимизация"
Самостоятельная работа
Дополнительные материалы
Тест для самопроверки
Задания на программирование
1.5.5. Построение графиков с помощью seaborn. Работа с Git
Лекция
Построение графиков в seaborn
Создание виртуального окружения
Инструменты статического анализа кода
Git. Работа с распределенными системами управления версиями
Практическая работа
Семинар "Seaborn и практика по деревьям"

Самостоятельная работа
Дополнительные материалы
Тест для самопроверки
Задания на программирование

Семестр: 2 (Весенний)

2. Основы машинного обучения

2.1. Математические основы машинного обучения

2.1.1. Введение в машинное обучение. Метод ближайших соседей

Лекция

Введение. Сферы применения машинного обучения

Введение в машинное обучение

Инструкция по настройке локальной машиныФайл

Инструкция по работе с различными онлайн-средами для ноутбуковФайл

Основные понятия машинного обучения

Формальная задача машинного обучения с учителем

Метод k ближайших соседей

Метрики классификации

Реализация kNN в Python

Практическая работа

Семинар "Метод ближайших соседей"

Самостоятельная работа

Дополнительные материалы

Задания на программирование

2.1.2. Случайность. Наивный Байесовский классификатор

Лекция

Вероятность. Свойства вероятности

Условная вероятность. Теорема Байеса

Наивный Байесовский классификатор

Реализация наивного байесовского классификатора

Эмпирические функции распределения

Практическая работа

Семинар "Наивный Байесовский классификатор"

Самостоятельная работа

Дополнительные материалы

Задания на программирование

2.1.3. Оптимизация

Лекция

Производная и ее применения

Градиентная оптимизация

Условная оптимизация

Решение задачи оптимизации градиентными методами

Практическая работа

Семинар "Оптимизация"

Самостоятельная работа

Дополнительные материалы

Задания на программирование

2.1.4. Задача линейной классификации. Логистическая регрессия

Лекция

Задача линейной классификации

Правдоподобие

Логистическая регрессия
Мультиклассовая классификация
Практическая работа
Семинар "Логистическая регрессия"
Самостоятельная работа
Дополнительные материалы
Задания на программирование
2.1.5. Метод опорных векторов
Лекция
Метод опорных векторов
Нелинейный метод опорных векторов
Практическая работа
Выполнение задачи на программирование по теме урока
Самостоятельная работа
Дополнительные материалы
Задания на программирование
Проект ML Pipeline
2.2. Машинное обучение
2.2.1. Решающие деревья
Лекция
Решающие деревья
Процедура построения дерева решений
Критерии информативности: Энтропия
Критерий Джини
Критерии в задаче регрессии. Усечение деревьев
Специфические свойства деревьев
Практика по деревьям
Практическая работа
Выполнение задачи на программирование по теме урока
Самостоятельная работа
Дополнительные материалы
Задания на программирование
2.2.2. Случайный лес
Лекция
Техника ансамблирования
RSM
Дилемма смещения
Смешивание
Стекинг
Ансамбли деревьев
Практическая работа
Выполнение задачи на программирование по теме урока
Самостоятельная работа
Дополнительные материалы
Задания на программирование
2.2.3. Оценка качества классификации. Оценка значимости признаков
Лекция
Оценка качества классификации
Методы кросс-валидации
Cross-validation riddle
Оценка значимости признаков
Feature importances
Практическая работа
Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Задания на программирование

2.2.4. Градиентный бустинг

Лекция

Бустинг

Градиентный бустинг

Визуализация градиентного бустинга

CatBoost

Сравнение градиентного бустинга с другими ансамблевыми методами

Реализация градиентного бустинга в Python

Практическая работа

Выполнение задачи на программирование по теме урока

Самостоятельная работа

Дополнительные материалы

Задания на программирование

2.2.5. Машинное обучение без учителя. Кластеризация

Лекция

Обучение без учителя. Кластеризация

Методы понижения размерности

Рекомендательные системы

Введение в кластеризацию

Разнообразие задач кластеризации

Практическая работа

Выполнение задачи на программирование по теме урока

Самостоятельная работа

Дополнительные материалы

Задания на программирование

2.2.6. Алгоритмы кластеризации

Лекция

K-Means

ЕМ-алгоритм

Агломеративная иерархическая кластеризация

Графы и методы на основе плотности точек

Алгоритм DBSCAN

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Задания на программирование

2.2.7. Выбор метода кластеризации. Оценка качества кластеризации

Лекция

Выбор метода кластеризации

Оценка качества и рекомендации

Практическая работа

Выполнение задачи на программирование по теме урока

Самостоятельная работа

Дополнительные материалы

Задания на программирование

2.3. Глубокое обучение

2.3.1. Введение в глубокое обучение

Лекция

История искусственных нейронных сетей

Нейронные сети

Механизм обратного распространения ошибки
Функции активации
Интерактивное демо
Нейронные сети. Итоги
Простейшая нейросеть на PyTorch
Практическая работа
Выполнение задачи на программирование по теме урока
Самостоятельная работа
Дополнительные материалы
Задания на программирование
2.3.2. SGD доработки
Лекция
SGD доработки
Регуляризация в DL
Проблема переобучения
Аугментация и итоги
PyTorch: модель с регуляризацией
Практическая работа
Выполнение задачи на программирование по теме урока
Самостоятельная работа
Дополнительные материалы
Задания на программирование
2.3.3. Векторные представления слов
Лекция
Введение в NLP
Предварительная обработка текста
Извлечение признаков
Векторное представление слов (Word Embeddings)
Векторные представления слов. Визуализация. (Word embeddings visualization)
Визуализация в Python
Визуализация в Python на примере векторных представлений слов
Практическая работа
Выполнение задачи на программирование по теме урока
Самостоятельная работа
Дополнительные материалы
Проект "Генерация поэзии"
2.3.4. Рекуррентные нейронные сети. Проблема затухающего градиента
Лекция
Языковое моделирование
Рекуррентные нейронные сети
RNN
LSTM
Проблема затухающих градиентов
Проблема взрывающихся градиентов
Языковое моделирование: реализация в Python
Практическая работа
Выполнение заданий по теме лекции
Самостоятельная работа
Дополнительные материалы
Задания на программирование
2.3.5. Обработка изображений. Сверточные нейронные сети.
Лекция
Сверточные слои
Интерактивная демонстрация

Padding, Strides, Pooling
Обзор архитектур
Свертки для изображений. Базовый обзор, примеры
Практическая работа
Выполнение задачи на программирование по теме урока
Самостоятельная работа
Дополнительные материалы
Задания на программирование

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Система дистанционного обучения:

Обучающемуся необходимо наличие доступа в сеть интернет, компьютер.

Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

6.Перечень рекомендуемой литературы

Основная литература

1. Python и анализ данных, Электрон. версия печ. публикации / У. Маккини. — Москва, ДМК Пресс, 2020
2. Python и анализ данных, Первичная обработка данных с применением pandas, NumPy и IPython / У. Маккини. — Москва, ДМК Пресс, 2020.— URL: <https://e.lanbook.com/book/131721> (дата обращения: 26.01.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)
3. Курс дифференциального и интегрального исчисления : в 3 т. Т. 1 : учебник для вузов : рек. М-вом образования Рос. Федерации / Г. М. Фихтенгольц ; пред. и прим. А. А. Флоринского .— 8-е изд. / .— М. : Физматлит, 2001, 2003, 2006, 2007 .— 680 с.
4. Лекции по математическому анализу. В 3 частях, Часть 1, Функции одной переменной, учебник для вузов/Я. М. Дымарский , -Москва, МФТИ, 2020
5. Аналитическая геометрия и линейная алгебра [Текст] : в 2 ч. : учеб. пособие для вузов. Ч. 1 / А. Е. Умнов ; М-во образования и науки Рос. Федерации, Моск. физико-техн. ин-т (гос. ун-т) .— 2-е изд., испр. и доп. — М. : Изд-во МФТИ, 2006 .— 272 с.

Литература из средств кафедр:

1. Машинное обучение. Паттерны проектирования: Пер. с англ./ В. Лакшманан, С. Робинсон, М. Мунн. - СПб.: БХВ-Петербург, 2022. - 448 с.
2. Практическая статистика для специалистов Data Science: Пер. с англ. / П. Брюс, Э. Брюс, П. Гедек. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2021. — 352 с.
3. Простой Python. Современный стиль программирования. Б. Любанович. - СПб., Питер, 2018

Дополнительная литература

1. Введение в теорию вероятностей и ее приложения [Текст] : в 2 т : учеб. пособие для вузов. Т. 1 / В.Феллер ; пер. с пересмотр. 3-го англ. изд. Ю. В. Прохорова ; [придесл. А. Н. Колмогорова] .— М. : Мир, 1984 .— 528 с.
2. Курс аналитической геометрии и линейной алгебры [Текст] : учебник для вузов / Д. В. Беклемишев .— 12-е изд., испр. — М. : Физматлит, 2008, 2009 .— 312 с.
3. Python и машинное обучение [Текст], крайне необходимое издание по новейшей предсказательной аналитике для более глубокого понимания методологии машинного обучения/С. Рашка, -М., ДМК Пресс, 2017

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. Официальный сайт Python - <https://www.python.org/>
2. Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>
3. Среда разработки IDE PyCharm - <https://www.jetbrains.com/pycharm/>
4. PEP 8 – Style Guide for Python Code - <https://peps.python.org/pep-0008/>
5. Официальная документация (англ.): <https://docs.python.org/3/>
6. Документация по исключениям: <https://docs.python.org/3/library/exceptions.html>
7. Модель GitFlow: <https://nvie.com/posts/a-successful-git-branching-model/>
8. Современный подход к версионированию для CI/CD:
<https://www.atlassian.com/continuous-delivery/continuous-integration/trunk-based-development>
9. Курс «Программирование на Python»: stepik.org/course/67/
10. Игровой тренажер по Python: py.checkio.org
11. NumPy documentation — NumPy v1.24 Manual <https://numpy.org/doc/stable/>
12. Audio and Digital Signal Processing(DSP) in Python (pythonforengineers.com)
<https://new.pythonforengineers.com/blog/audio-and-digital-signal-processingdsp-in-python/>
13. MP3 to Numpy and back | Kaggle
<https://www.kaggle.com/code/gabrielmilan/mp3-to-numpy-and-back>
14. GitHub: Let's build from here · GitHub <https://github.com/>
15. Harvard Dataverse <https://dataverse.harvard.edu/>
16. User Guide — pandas 2.0.2 documentation ([pydata.org](https://pandas.pydata.org))
17. Pandas Pivot Table Explained - Practical Business Python (pbpython.com)
https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html
18. Моем датасет: руководство по очистке данных в Python (proglab.io)
<https://proglab.io/p/moem-dataset-rukovodstvo-po-ochistke-dannyh-v-python-2020-03-27>
19. pandas documentation — pandas 1.5.3 documentation <https://pandas.pydata.org/docs/>
20. Comparison with other tools — pandas 2.0.2 documentation ([pydata.org](https://pandas.pydata.org))
https://pandas.pydata.org/docs/getting_started/comparison/index.html
21. Comprehensive Guide to Grouping and Aggregating with Pandas - Practical Business Python (pbpython.com) <https://pbpython.com/groupby-agg.html>
22. Наглядная шпаргалка по операциям с DataFrame в pandas для data wrangling и не только (tproger.ru) <https://tproger.ru/articles/pandas-data-wrangling-cheatsheet>
23. Statistical functions (scipy.stats) — SciPy v1.10.1 Manual
<https://docs.scipy.org/doc/scipy/reference/stats.html>
24. Understanding Confidence Intervals | Easy Examples & Formulas ([scribbr.com](https://www.scribbr.com))
<https://www.scribbr.com/statistics/confidence-interval/>
25. 6 Steps to Evaluate a Statistical Hypothesis Testing (enago.com)
<https://www.enago.com/academy/evaluate-statistical-hypothesis-testing/>
26. 17 Statistical Hypothesis Tests in Python (Cheat Sheet) - MachineLearningMastery.com
https://machinelearningmastery.com/statistical-hypothesis-tests-in-python-cheat-sheet/?__cf_chl=tk=k9U5fBCvimOWBbBLF2DKPZToD43bZGKm20Yq_lvEfWc-1707807813-0-4453
27. Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных –
<http://www.machinelearning.ru/>
28. Python Numpy Tutorial - <https://cs231n.github.io/python-numpy-tutorial/>
29. Learn Git Branching - https://learngitbranching.js.org/?locale=ru_RU
30. Deep Learning (deeplearningbook.org) <https://www.deeplearningbook.org/>
31. Mathematics for Machine Learning (mml-book.github.io)
<https://mml-book.github.io/book/mml-book.pdf>
32. Unbiased Research <https://unbiasedresearch.blogspot.com/>
33. Введение в машинное обучение с помощью scikit-learn (перевод документации) / Хабр (habr.com) <https://habr.com/ru/articles/264241/>
34. Supervised learning — scikit-learn 1.3.0 documentation
https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
35. Кластеризуем лучше, чем «метод локтя» / Хабр (habr.com)
<https://habr.com/ru/companies/jetinfosystems/articles/467745/>
36. Кластеризация - scikit-learn <https://scikit-learn.ru/clustering/>

- 37. 10 библиотек Python для машинного обучения и искусственного интеллекта - UPROGER | Программирование <https://uproger.com/10-bibliotek-python-dlya-mashinnogo-obucheniya/?ref=vc.ru>
- 38. TensorFlow <https://www.tensorflow.org/?hl=ru>
- 39. Keras: Deep Learning for humans <https://keras.io/>
- 40. PyTorch <https://pytorch.org/>
- 41. Caffe | Deep Learning Framework (berkeleyvision.org) <https://caffe.berkeleyvision.org/>

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Документация Postgres про сравнение строк - <https://postgrespro.ru/docs/postgrespro/9.5/functions-matching>
Документация Postgres про другие функции работы со строками - <https://postgrespro.ru/docs/postgrespro/9.5/functions-string>
Тестер регулярных выражений - <https://www.regextester.com>
Интерактивный учебник по SQL - <http://www.sql-tutorial.ru/ru/content.html>
Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>
Начало работы с Power BI - <https://docs.microsoft.com/ru-ru/power-bi/fundamentals/desktop-getting-started>
Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных – <http://www.machinelearning.ru/>
Информационная система «Единое окно доступа к образовательным ресурсам» (ИС «Единое окно») – <http://window.edu.ru/>
Платформа открытое образование – <https://openedu.ru/>

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Самостоятельная работа подразделяется на аудиторную и внеаудиторную. Аудиторную самостоятельную работу составляют практические задания, которые выполняются слушателями во время учебных занятий, результаты ее выполнения проверяются и оцениваются преподавателем в учебном процессе.

Внеаудиторная самостоятельная работа включает формы: изучение дополнительной литературы, подготовка итоговых проектов по модулям, подготовка проекта.

Основными критериями качества организации самостоятельной работы служит наличие контроля результатов самостоятельной работы.

Основными современными формами организации самостоятельной работы являются творческие работы и работа с информационными компьютерными технологиями.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

| | |
|----------------------------|---|
| по направлению: | Прикладные математика и физика |
| профиль подготовки: | Природоподобные технологии и биомиметический дизайн материалов и систем Физтех-школа природоподобных, плазменных и ядерных технологий им. И.В. Курчатова центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" |
| курс: | <u>1</u> |
| квалификация: | магистр |

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Зачет
- 2 (весенний) - Зачет

Разработчики:

Р.Г. Нейчев, старший преподаватель
А.П. Чистов, ассистент
М.А. Певцова, методист
Ж.И. Зубцова, канд. физ.-мат. наук, эксперт

1. Компетенции, формируемые в процессе изучения дисциплины

| Код и наименование компетенции | Индикаторы достижения компетенции |
|--|---|
| УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной задачи | УК-3.1 Организует и координирует работу участников проекта, способствует конструктивному преодолению возникающих разногласий и конфликтов |
| | УК-3.4 Способен планировать командную работу, распределять поручения членам команды, организовать обсуждение разных идей и мнений |
| ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий | ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов |

2. Показатели оценивания компетенций

В результате изучения дисциплины «Основы машинного и глубокого обучения» обучающийся должен:

знать:

- принципы организации кода;
- базовые типы и конструкции языка Python;
- понятия вектора и матрицы, векторного пространства, нормы вектора, ортогональности и гиперплоскости;
- как выполнять базовые операции над векторами и матрицами;
- как библиотека NumPy помогает в научных вычислениях и обработке данных;
- назначение и принцип работы библиотеки NumPy;
- основные методы обработки данных, упрощающие решение задач машинного обучения;
- в каких сферах применяется машинное обучение;
- основные понятия машинного обучения: датасет (выборка), объект, признак, таргет, матрица объект-признак, машинное обучение с учителем, таргет, модель, предсказание, функция потерь, параметр, гиперпараметр;
- формальную постановку задачи машинного обучения с учителем;
- как строится сверточная нейронная сеть;
- часто используемые в CNN техники (padding, striding, pooling);
- различные архитектуры, использующиеся в машинном обучении;
- понятие коллекции;
- назначение коллекций в разработке;
- в каких задачах машинного обучения используются линейные модели;
- теорему Гаусса-Маркова;
- основные понятия теории вероятностей;
- понятие условной вероятности, дискретных и непрерывных случайных величин;
- центральную предельную теорему и теорему Байеса;
- в каких задачах можно применить наивный Байесовский классификатор;
- как строится рекуррентная нейронная сеть (RNN);
- разновидностями архитектуры RNN — LSTM и GRU;
- проблемы затухающих и взрывающихся градиентов;
- понятие и роль специальных методов классов в программировании на Python;
- способы подключения модулей стандартной библиотеки;
- модули стандартной библиотеки;
- виды графического представления данных и ситуации их использования;
- возможности библиотеки matplotlib для построения различных видов графиков и настройки их отображения;
- функции и их свойства;
- как находить производные функции, что такое экстремумы и критические точки функции;
- что такое градиент функции;
- методы оптимизации;
- задачу обработки естественного языка;
- понятия правдоподобия в задачах машинного обучения;
- различные метрики оценки качества классификации;
- основные подходы градиентной оптимизации;
- основные техники регуляризации в глубоком обучении;
- метод опорных векторов, используемый для задач классификации и регрессионного анализа;
- способы создания нелинейного классификатора с помощью так называемого ядерного трюка (kernel trick);
- критерии информативности: энтропия и критерий Джини;
- как использовать решающие деревья в задаче регрессии;
- основные нелинейные функции активации;
- механизмы обратного распространения ошибки;
- принципы работы с функциями: синтаксис функций, способы задания аргументов и возврата значений;
- понятия классов и объектов в Python, их взаимосвязь;
- особенности объектно-ориентированной модели в Python;
- понятие и механизмы наследования, его роль в программировании;
- подходы к обработке ошибок;
- механизм формирования исключений;
- возможности библиотеки seaborn для построения различных видов графиков;
- основы математической статистики;
- различные техники ансамблирования и теоретические предпосылки к их применению;
- процедуру bootstrap, на основе которой строится метод ансамблирования бэггинг;
- принципы построения случайного леса;
- как использовать технику блендинга, которая позволяет строить ансамбли не параллельно друг другу, а последовательно;

уметь:

- Выполнять практические задачи и проекты в команде;
- Установить интерпретатор Python на компьютер;
- Установить среду разработки PyCharm;
- Написать простой код на Python;
- использовать базовые типы и конструкции Python для написания простых программ;
- применять NumPy для работы с векторами и матрицами;
- применять метод kNN для решения задач машинного обучения;
- обучать сверточную нейронную сеть (CNN);
- использовать сверточную нейронную сеть для обработки изображений;
- использовать встроенные в Python коллекции для написания программ;
- сопоставлять и выбрать необходимую структуру данных для конкретной практической задачи;
- формально поставить задачу линейной регрессии;
- использовать L1- и L2-регуляризации для решения задач машинного обучения;
- использовать специальные методы в написании программ;
- применять функции для построения основных видов графиков и настраивать внешний вид графиков (цвет, подписи, легенда, сетка);
- строить 3D-изображения с помощью библиотек Python и использовать их для задач компьютерного зрения;
- решать задачи оптимизации градиентными методами;
- предобрабатывать текстовые данные и извлекать из них признаки;
- строить информативные векторные представления слов;
- решать задачи линейной классификации в машинном обучении;
- использовать модель логистической регрессии в задачах бинарной и мультиклассовой классификации;
- бороться с переобучением нейросети с помощью регуляризации;
- строить модель с регуляризацией в PyTorch;
- использовать решающие деревья в задачах машинного обучения;
- описывать линейные и нелинейные зависимости с помощью нейросетей;
- строить простейшую нейросеть на PyTorch;
- использовать функции как объект, функции высших порядков, лямбда-функции;
- выполнить чтение и запись в файл;
- применять рекомендации PEP 8 для написания кода;
- создавать классы и использовать методы классов;
- проектировать необходимые классы и методы классов;
- проектировать классы с использованием механизма наследования;
- использовать конструкции языка для генерации исключений на Python;
- создавать виртуальное окружение;
- выполнять статистический анализ программного кода;
- работать с распределенной системой контроля версий Git;
- получить информацию о DataFrame, вычислить описательные статистики для числовых данных, обратиться к элементам DataFrame по индексу и порядковому номеру, изменить индекс;
- выполнять поиск, фильтрацию и сортировку DataFrame с применением методов библиотеки Pandas;
- вычислять статистику по признакам, применять функции к данным, рассчитывать новые значения;
- работать с несколькими таблицами с помощью инструментов библиотеки Pandas;
- строить графики в Pandas;
- подбирать методы, признаки и число кластеров для кластеризации;
- оценивать влияние признаков на прогноз модели машинного обучения с помощью библиотеки SHAP;
- использовать метод кросс-валидации для оценки качества модели;
- пользоваться EM-алгоритмом;
- пользоваться методом DBSCAN;
- решать задачи классификации и регрессии с помощью градиентного бустинга;
- использовать градиентный бустинг;
- эффективно использовать большие объемы данных посредством понижения их размерности.

владеть:

- стандартными структурами данных в Python, умением писать функции на Python, применять функциональные особенности языка, работать с файлами с помощью языка Python;
- механизмами наследования, создавать классы и работать с ними, обрабатывать исключения;
- навыками выбора подходящего метода оптимизации для конкретной задачи;
- навыками применения библиотеки Python для построения модели линейной регрессии, решающих деревьев и композиций алгоритмов, для обучения метрических алгоритмов, SVM, байесовских моделей.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Задание на программирование

1. Приветствие

На вход программа получает строку — имя пользователя. Считайте её и выведите на экран фразу: «Привет, [имя]!».

Обратите внимание, что выводимая строка должна быть написана в точности так, как в задании. Если будет отличаться регистр букв или будут отсутствовать знаки препинания, программа засчитана не будет.

| | |
|----------|-------------------|
| Ввод | Вывод |
| Валентин | Привет, Валентин! |

2. Вычисление значения предыдущего числа и следующего

На вход программа получает целое число a . Получите значения предыдущего числа и следующего. Выведите результат $a-1$, a , $a+1$ по одному числу в строку.

| | |
|------|-----------|
| Ввод | Результат |
| 5 | 4 |
| | 5 |
| | 6 |

3. Питомец

Егор разрабатывает приложение «Кошечки-собачки». Ему нужно считать данные о питомце, которые введены в консоль, и подтвердить их ввод путем вывода на экран.

На вход программа получает четыре параметра: имя питомца (name), вид животного (animal), порода (breed) и возраст (age).

Необходимо вывести введенную информацию в виде:

Ваш питомец: animal name
 Порода питомца: breed
 Возраст питомца: age

| | |
|----------|--------------------------|
| Ввод | Результат |
| Кузя | Ваш питомец: кот Кузя |
| кот | |
| мейн-кун | Порода питомца: мейн-кун |
| 5 | Возраст питомца: 5 |

4. Месяц и время года

На вход программы поступает целое число — порядковый номер месяца. Напишите программу, которая выводит построчно название месяца и время года, к которому этот месяц относится. Название месяца и времени года должны выводиться с маленькой буквы.

Если вводимое число не попадает в диапазон $[1; 12]$, программа выводит “ошибка”.

| | |
|------|-------|
| Ввод | Вывод |
| 7 | июль |

5. Сумма ряда

Python часто используют для решения математических задач, одна из них — вычисление суммы ряда. Напишите программу, которая будет вычислять и выводить на экран целое число — сумму квадратов натуральных чисел до N , где N — натуральное число ($N > 1$), подаваемое на вход программы.

$\backslash (1+2^2+3^2+\dots+N^2 \backslash)$

Ввод Результат

20 2870

6. Минимальное число

Программа считывает натуральное число. По данному натуральному числу найдите минимальное натуральное число, состоящее из тех же цифр, что и данное, и выведите его на экран.

Важно! Число не может начинаться с 0.

Ввод Вывод

51 15

121 112

809 809

7063 3067

7. Точки с массой

Даны N точек на прямой. У каждой известны координата X и масса M . Задача: вывести координаты точек в порядке убывания массы.

Программа считывает натуральное число N — количество точек. Далее программа построчно считывает параметры для каждой n -ой точки: координату x_i и массу m_i . Оба параметра являются целыми числами, масса является положительным числом. Гарантируется, что строгого совпадения масс двух точек в считываемых данных не встречается.

Программа должна вывести координаты x_i точек, отсортированных по убыванию массы, каждая в новой строке.

Ввод Результат

5 1

-1 1 2

7 3 7

5 2 5

1 5 -1

2 4

8. Доменные имена

Программа считывает URL-адрес сайта (например, <https://vk.com>). Определите, находится ли сайт в международном домене, и если нет, к какой стране он относится.

Если домен является международным, выведите на экран слово «Международный». Если домен является национальным, выведите на экран название страны.

Варианты доменов, которые могут быть на входе:

- .uk Великобритания
- .de Германия
- .il Израиль
- .in Индия
- .kz Казахстан
- .mm Мьянма
- .om Оман

- .ru Россия
- .uz Узбекистан
- .et Эфиопия
- .com Международный
- .org Международный
- .net Международный

Ввод Результат
<https://vk.com> Международный

9. Числа Фибоначчи

Напишите функцию `fibonacci(index)`, которая находит число Фибоначчи с нужным номером. Аргумент функции `index` — целое положительное число.

Последовательность Фибоначчи задается формулой:

$$F(1) = 0$$

$$F(2) = 1$$

$$F(N) = F(N - 1) + F(N - 2) \text{ для } N \geq 3$$

То есть, каждое последующее число равно сумме двух предыдущих чисел. Если требуется больше информации о данной последовательности, вы можете обратиться, например, к источнику.

| Аргументы | Возвращаемое значение |
|-----------|-----------------------|
|-----------|-----------------------|

| | |
|---|---|
| 1 | 0 |
|---|---|

| | |
|---|---|
| 2 | 1 |
|---|---|

| | |
|----|----|
| 10 | 34 |
|----|----|

10. Последний разряд

Напишите функцию `last_discharge(a)`, которая в качестве аргумента получает строку, содержащую число `a`, и возвращает это же число, уменьшенное на единицу в последнем его разряде в виде строки.

Обратите внимание:

- в последнем разряде может быть ноль,
- входная строка может представлять как целое число, так и число с плавающей точкой,
- входная строка содержит число строго больше нуля.

| Аргументы | Возвращаемое значение |
|-----------|-----------------------|
|-----------|-----------------------|

| | |
|----|----|
| 42 | 41 |
|----|----|

| | |
|-------|-------|
| 45.12 | 45.11 |
|-------|-------|

| | |
|------|------|
| 1.00 | 0.99 |
|------|------|

| | |
|-----|-----|
| 1.0 | 0.9 |
|-----|-----|

Введение в машинное обучение. Метод ближайших соседей

11. Подсчет расстояний для kNN

В рамках этой задачи необходимо заполнить файл `k_nearest_neighbor.py` в папке для решения соответствующего ноутбука. Вы можете открыть его локально или же воспользоваться ссылкой на Colab в папке выше.

После выполнения всех шагов в ноутбуке и прохождения локальных тестов, сохраните локально и отправьте получившийся .py файл сюда.

Внимание! Вердикт ОК означает, что ваш код запустился.

Методы оптимизации и регрессионного анализа

12. Вычисление производных

В рамках этой задачи необходимо заполнить файл `'derivatives.py'` в папке. и отправить получившийся .py файл сюда.

Внимание! Вердикт ОК означает, что ваш код запустился.

Решение должно работать в python 3.5 и не использовать никаких внешних библиотек кроме numpy и scipy.

Метод опорных векторов. Оценка качества классификации. Методы кросс-валидации

13. Ядра для SVM

В рамках этой задачи необходимо заполнить файл svm.py по ссылке и отправить получившийся .py файл сюда.

Внимание! Вердикт ОК означает, что ваш код запустился.

Решение должно работать в python 3.5 и не использовать никаких внешних библиотек кроме numpy, scipy и PyTorch.

Случайность. Наивный Байесовский классификатор

14. Распределение Лапласа

В рамках этой задачи необходимо заполнить файл distribution.py в папке для решения соответствующего ноутбука. Вы можете открыть его локально или же воспользоваться ссылкой на Colab в папке выше.

После выполнения всех шагов в ноутбуке и прохождения локальных тестов, сохраните локально и отправьте получившийся .py файл.

Внимание! Вердикт ОК означает, что ваш код запустился.

Решение должно работать в python 3.9 и не использовать никаких внешних библиотек кроме numpy. Использование готового распределения из scipy запрещено!

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Тестовые задания

1. Что такое инструкция в языках программирования?

- команда для выполнения какого-либо действия
- документ с перечислением всех команд языка
- данные, которые пользователь вводит с консоли в программу
- спецсимволы, используемые для написания кода

2. Какие элементы используют для формирования структуры программы в Python? Выберите все верные варианты.

- пробелы
- двоеточие
- табуляция
- фигурные скобки
- команды языка
- знак #

3. Что будет выведено в консоль после выполнения кода?

```
mylist = ['P', 'y', 't', 'h', 'o', 'n']  
print (mylist [1], mylist [-1])
```

- y n
- P n
- ошибка
- y P
- P y

4. Какая структура данных будет создана после выполнения кода?

```
fruits = {'apple':'яблоко', 'banana':'банан', 'lemon':'лимон', 'orange':'апельсин', 'pear':'груша'}
```

- словарь
- множество
- массив

- кортеж

5. Что будет выведено в консоль после выполнения кода?

```
a = lambda x, y: return x + y  
print (a(20, 15))
```

- ошибка
- 35
- 35.0
- None

6. Как нужно записать команду открытия файла 'new.txt' для записи, чтобы при этом первоначальные данные файла сохранились, а новые были добавлены в конец файла?

- open('new.txt', 'a')
- file.open('new.txt')
- open('new.txt', 'w')
- write('new.txt')

7. Какое определение верно описывает принцип полиморфизма?

- Одинаковый интерфейс с различной реализацией
- Отделение данных от кода, который их обрабатывает
- Объединение данных и кода, который их обрабатывает
- Разные методы с одинаковой реализацией

8. В какой строке программы допущена ошибка?

```
1 import abc  
2 class AbstractFigure:  
3  
4     @abc.abstractmethod  
5     def draw(self, screen):  
6         ...
```

9. Какие типы исключений может вызвать следующий код?

```
x = 1 / float(input())
```

- ZeroDivisionError
- ValueError
- InputError
- OverflowError

10. Сколько аргументов принимает конструктор класса Exception?

- любое количество
- один
- не более двух
- ни одного

11. Предположение i.i.d (независимые одинаково распределенные) обычно применяется к ...

наблюдениям

признакам

значениям целевой переменной

12. В задаче классификации целевая переменная ...

принимает только целочисленные значения

является вектором из действительных чисел

принимает значение из конечного множества ответов

13. Число соседей в методе kNN является ...
 Параметром
 Гиперпараметром
 Метрикой
 Функцией потерь
14. Функция $f(x)$ называется _____ для всех x_0 из области определения, если $(\lim)_{x \rightarrow x_0} [f(x) = f(x_0)]$
 Убывающей
 Непрерывной
 Выпуклой
 Возрастающей
15. Чем больше норма вектора, тем _____ будет функция расти в направлении градиента.
 Быстрее
 Медленнее
16. Теорема Гаусса-Маркова утверждает, что ...
 Минимизация MAE приводит к наилучшей линейной несмещённой оценке\
 Регуляризация приводит к более стабильному решению задачи регрессии
 Минимизация MSE приводит к наилучшей линейной несмещённой оценке
 Ошибки нескоррелированы в задаче регрессии
17. Использование SGD с Momentum позволяет...
 ускорить сходимость в некоторых случаях.
 избежать некоторых локальных минимумов.
 избежать седловых точек.
 получить аналитическое решение промежуточной оптимизационной задачи.
18. Нейронные сети могут быть представлены в виде комбинации...
 решающих деревьев и линейных функций.
 линейных функций и нелинейных активаций.
 решающих деревьев и нелинейных активаций.
19. Нейронные сети чаще всего обучаются с использованием механизма ...
 жадной оптимизации
 генетических алгоритмов
 прямой активации выхода
 обратного распространения ошибки

Критерии оценивания

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

Оценка «зачтено» выставляется студенту, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в зачет представлены ниже:

| | |
|--------------------|------------------|
| Модуль | Вклад в зачет, % |
| Оценка за модуль 1 | 40 |
| Оценка за модуль 2 | 40 |
| Зачет | 20 |

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины.