

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

**УТВЕРЖДЕНО**

**Директор по цифровизации  
образования**

**Д.И. Гриц**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Продвинутые методы машинного обучения
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Программная инженерия передовая инженерная школа радиолокации, радионавигации и программной инженерии центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	1
<b>квалификация:</b>	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  
Г.К. Тарасенко, преподаватель  
Н.А. Долгополов, преподаватель  
Ж.И. Зубцова, канд. физ.-мат. наук, эксперт  
Л.С. Орлова, методист

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и онлайн-образования "Пуск" 28.02.2024

## Аннотация

Дисциплина состоит из двух модулей:

Модуль 1. Анализ данных в Python и введение в машинное обучение

Модуль 2. Машинное обучение

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

- формирование/совершенствование компетенций слушателей в области решения профессиональных задач по работе с данными с помощью как основных, так и продвинутых методов машинного обучения, а также методов глубокого обучения.

#### Задачи дисциплины

- научиться использовать библиотеки Python для работы с данными;
- сформировать умение наглядно представлять результаты работы с данными;
- научиться решать простые задачи машинного обучения с учителем и без учителя;
- сформировать умение решать прикладные задачи машинного обучения;
- научиться решать прикладные задачи глубокого 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 цели	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, организовывать обсуждение разных идей и мнений
ОПК-3 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области информатики и вычислительной техники, учитывая особенности и ограничения различных методов решения	ОПК-3.7 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- возможности основных библиотек, используемых для анализа данных;
- как библиотека NumPy помогает в научных вычислениях и обработке данных;
- понятия вектора и матрицы, векторного пространства, нормы вектора, ортогональности и гиперплоскости;
- как выполнять базовые операции над векторами и матрицами;
- 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;
- 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;
- виды, источники и способы хранения данных (csv, tsv-файлы и другие);
- структуры данных и инструменты, предоставляемые библиотекой Pandas для работы с данными;
- терминологию, используемую в машинном обучении;
- виды линейных моделей обучения, метрики измерения качества линейных моделей;
- базовые сведения об ансамблевых моделях;
- производительность ансамблевых моделей;
- в каких задачах машинного обучения используются линейные модели;
- теорему Гаусса-Маркова;
- что такое градиент функции;
- методы оптимизации;
- понятия правдоподобия в задачах машинного обучения;
- как использовать модель логистической регрессии в задачах бинарной и мультиклассовой классификации;
- различные метрики оценки качества классификации;
- когнитивные законы и принципы восприятия информации человеком;
- виды графического представления данных и ситуации их использования;
- методы кластеризации и методы понижения размерности, принципы построения рекомендательных систем;
- метод опорных векторов, используемый для задач классификации и регрессионного анализа;
- способ создания нелинейного классификатора с помощью так называемого ядерного трюка (kernel trick);
- метрики оценки качества классификации;
- ROC-AUC;
- архитектуру Transformer;
- принцип позиционного кодирования;
- как работает декодер в Transformer;
- модель ELMo;
- модель BERT;
- задачу языкового моделирования;
- свойства нормального распределения;
- как использовать библиотеки Python для анализа данных в некоторых задачах машинного обучения;
- терминологию, используемую в нейросетях;
- архитектуры нейронных сетей;
- основные библиотеки Python, используемые для работы с нейросетями;
- возможности библиотеки matplotlib для построения различных видов графиков и настройки их отображения;
- об инструментах для визуализации данных, используемых в работе аналитика;
- функциональные возможности сервиса Yandex DataLens для построения и настройки графиков;
- критерии информативности: энтропию и критерий Джини;
- как использовать решающие деревья в задаче регрессии;
- что такое механизм внимания;
- особенности модели глубокого обучения Seq2Seq, и как ее используют в задаче машинного обучения;
- Self-Attention;
- Multi-Head Attention;
- различные техники ансамблирования и теоретические предпосылки к их применению;

уметь:

- использовать базовые операции по работе с массивами, математические и статистические функции библиотеки NumPy для решения прикладных задач;
- решать системы линейных уравнений в Python матричным методом, решать задачи с помощью системы линейных уравнений;
- применять NumPy для работы с векторами и матрицами;
- вычислять косинусную меру близости векторов и использовать ее для нахождения разницы между словами;
- применять метод kNN для решения задач машинного обучения;
- выполнять выгрузку данных с использованием библиотеки Pandas;
- проводить предварительную обработку данных с использованием библиотеки Pandas — получать информацию о DataFrame, работать со строками и столбцами;
- осуществлять группировку и агрегацию таблиц с использованием Pandas;
- строить модели линейной и логистической регрессии с использованием библиотек Python;
- рассчитывать метрику качества линейной и логистической регрессии;
- строить ансамблевые модели — решающее дерево, случайный лес, градиентный бустинг;
- формально поставить задачу линейной регрессии;
- использовать L1- и L2-регуляризации для решения задач машинного обучения;
- решать задачи оптимизации градиентными методами;
- решать задачи линейной классификации в машинном обучении;
- Выполнять практические задачи и проекты в команде;
- 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;
- реализовать методы кластерного анализа на примере искусственных данных, выполнить расчет оценок качества кластеризации с помощью библиотек Python;
- реализовать методы понижения размерности применительно к датасету с помощью Python;
- составить матрицу рейтингов с помощью Python и выполнять операции с ней;
- использовать метод кросс-валидации для оценки качества модели;
- использовать self attention для Transformer;
- кодировать энкодер Transformer;
- использовать метод t-SNE в задаче снижения размерности;
- отличать и понимать базовые статистические концепции — генеральная совокупность, выборка;
- вычислить точечные оценки и доверительные интервалы, интерпретировать их;
- проверять статистические гипотезы с использованием тестов на нормальность данных, равенство дисперсий, сравнение средних, взаимосвязь переменных;
- написать программу для вычисления результата сигмоидальной функции активации для заданных входных данных;
- написать код, реализующий свертку изображения и фильтра;
- реализовать простую нейросеть, состоящую из одного входного слоя, одного скрытого слоя и одного выходного слоя;
- 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;
- строить некоторые виды графиков в Yandex DataLens;
- строить графики и диаграммы с помощью библиотеки Altair, добавлять в графики интерактив и выполнять их настройку;
- строить статистические диаграммы с использованием библиотеки Seaborn;
- использовать решающие деревья в задачах машинного обучения;
- обучать сверточную нейронную сеть (CNN);
- использовать сверточную нейронную сеть для обработки изображений;
- создавать презентации на основе диаграмм с помощью движка Reveal.js, встраивая в него диаграммы из Yandex DataLens;
- создавать интерактивные инфопанели, применяя функционал библиотеки Altair;
- выбрать библиотеки Python для решения задачи анализа данных;
- выполнять анализ данных из одного и нескольких источников с использованием языка Python;
- строить и анализировать матрицу корреляции на Python;

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	Анализ данных в Python и введение в машинное обучение	15	30		90
2	Машинное обучение	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. Анализ данных в Python и введение в машинное обучение

##### 1.1. Python для анализа данных

##### 1.1.1. Библиотека NumPy

Лекция

Введение в анализ данных и машинное обучение

Вычислительные функции библиотеки NumPy. Массивы

Векторы. Решение линейных уравнений

Использование NumPy в задачах обработки данных. Генерация мелодий

Работа с табличными данными и векторами

Работа с изображением с использованием NumPy

Практическая работа

Семинар "Библиотека NumPy"

Самостоятельная работа

Дополнительные материалы

Библиотека NumPy

Примеры работы с NumPy

Тест для самопроверки

Задания для самопроверки

##### 1.1.2. Получение и предобработка данных. Первичная работа с объектом DataFrame

Лекция

Виды и источники данных

Предобработка данных  
Первичная работа с датафреймом  
Введение в агрегирование и сводные таблицы  
Практическая работа  
Семинар "Получение и предобработка данных. Первичная работа с объектом DataFrame"  
Самостоятельная работа  
Дополнительные материалы  
Получение и предобработка данных  
Используемые наборы данных  
Тест для самопроверки  
Задания для самопроверки  
1.1.3. 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)  
Практическая работа  
Семинар "Описательная статистика. Анализ данных с помощью Pandas"  
Самостоятельная работа  
Дополнительные материалы  
Анализ данных с Pandas  
Тест для самопроверки  
Задания для самопроверки  
1.1.4. Статистика вывода  
Лекция  
Выборка и генеральная совокупность  
Распределения  
Оценки  
Тестирование гипотез  
Определение неисправностей в подшипниках через анализ экспериментальных данных  
Практическая работа  
Семинар "Статистика вывода"  
Самостоятельная работа  
Дополнительные материалы  
Статистика вывода  
Тестирование гипотез  
Определение неисправностей в подшипниках  
Используемые наборы данных  
Тест для самопроверки  
Задания для самопроверки  
1.2. Визуализация данных  
1.2.1. Введение в визуализацию. Библиотека matplotlib  
Лекция  
Как мы воспринимаем информацию  
Методы визуализации  
Библиотека Matplotlib  
3D-визуализация графики для машинного обучения  
Практическая работа  
Семинар "Введение в визуализацию. Библиотека matplotlib"  
Самостоятельная работа  
Дополнительные материалы  
Библиотека matplotlib  
Используемый набор данных

Примеры 3D-визуализации

Исходные данные для 3D-визуализации

Тест для самопроверки

Задания для самопроверки

### 1.2.2. Прикладные инструменты визуализации данных

Лекция

Yandex DataLens. построение диаграмм без программирования

Использование библиотек Python для визуализации. Библиотека Altair

Использование библиотек Python для визуализации. Библиотека Seaborn

Практическая работа

Семинар "Прикладные инструменты визуализации данных"

Самостоятельная работа

Дополнительные материалы

Библиотеки Altair, Altair вер. 5, Seaborn

Исходные данные для визуализацииФайл

Тест для самопроверки

Задания для самопроверки

### 1.2.3. Диаграммы в контексте: инфопанели и презентации

Лекция

Интерактивные средства и связанные представления

Презентации на основе диаграмм. Общие практики

Подготовка инфопанелей и презентаций с помощью Yandex DataLens

Подготовка инфопанелей и презентаций с помощью библиотеки Altair

Практическая работа

Семинар "Диаграммы в контексте: инфопанели и презентации"

Самостоятельная работа

Дополнительные материалы

Дашборды в Altair, в Altair вер. 5

Исходные данные для визуализацииФайл

Тест для самопроверки

Задания для самопроверки

Индивидуальное задание-проект по теме «Визуализация данных»

### 1.2.4. Примеры использования библиотек Python

Лекция

Пример анализа данных с помощью библиотеки Pandas

Пример анализа данных из нескольких источников

Работа с матрицей корреляции

Создание интерактивных графиков

Категоризация на примере анализа данных электроавтомобилей

Практическая работа

Семинар "Примеры использования библиотек Python"

Самостоятельная работа

Дополнительные материалы

Примеры использования библиотек Python

Исходные данные для примеров

Тест для самопроверки

Задания для самопроверки

### 1.3. Введение в машинное обучение

#### 1.3.1. Введение в линейную алгебру для машинного обучения

Лекция

Векторы. Основные операции над векторами

Матрицы. Основные операции над матрицами

Линейная алгебра в NumPy

Практическая работа



Семинар "Введение в линейную алгебру для машинного обучения"

Самостоятельная работа

Дополнительные материалы

Линейная алгебра для машинного обучения

Тест для самопроверки

Задания для самопроверки

1.3.2. Машинное обучение с учителем

Лекция

Введение в машинное обучение

Линейные модели

Измерение качества модели

Ансамблевые модели

Практическая работа

Семинар "Машинное обучение с учителем"

Самостоятельная работа

Дополнительные материалы

Машинное обучение с учителем

Тест для самопроверки

Задания для самопроверки

1.3.3. Машинное обучение без учителя

Лекция

Обучение без учителя. Кластеризация

Методы понижения размерности

Рекомендательные системы

Самостоятельная работа

Дополнительные материалы

Тест для самопроверки

Задания для самопроверки

1.3.4. Основы нейронных сетей

Лекция

Основы нейронных сетей

Архитектуры нейронных сетей

Практическая работа

Семинар "Машинное обучение без учителя и нейросети"

Самостоятельная работа

Дополнительные материалы

Тест для самопроверки

Задания для самопроверки

## Семестр: 2 (Весенний)

### 2. Машинное обучение

#### 2.1. Классическое обучение с учителем

##### 2.1.1. Введение в машинное обучение. Метод ближайших соседей

Лекция

Введение. Сферы применения машинного обучения

Инструкция по настройке локальной машины

Инструкция по работе с различными онлайн-средами для ноутбуков

Основные понятия машинного обучения

Формальная задача машинного обучения с учителем

Метод k ближайших соседей

Метрики классификации

Реализация kNN в Python

Практическая работа

Выполнение задания на программирование

Самостоятельная работа

Дополнительные материалы

Задание 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

Выполнение задачи на программирование по теме урока, тестирование

Самостоятельная работа

Дополнительные материалы

Задание на программирование

2.1.3. Методы оптимизации и регрессионного анализа

Лекция

Производная и ее применения

Градиентная оптимизация

Условная оптимизация

Решение задачи оптимизации градиентными методами

Линейные модели машинного обучения

Линейная регрессия

Теорема Гаусса-Маркова

L1 и L2 регуляризация

Решение линейной регрессии и анализ устойчивости решения

Практическая работа

Выполнение задачи на программирование по теме урока, тестирование

Самостоятельная работа

Дополнительные материалы

Задание на программирование

2.1.4. Задача линейной классификации. Логистическая регрессия

Лекция

Задача линейной классификации

Правдоподобие

Логистическая регрессия

Мультиклассовая классификация

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Задание на программирование

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

Лекция

Метод опорных векторов

Нелинейный метод опорных векторов

Оценка качества классификации

Методы кросс-валидации

Cross-validation riddle

Практическая работа

Выполнение задачи на программирование по теме урока, тестирование

Самостоятельная работа

Дополнительные материалы

Задание на программирование

#### 2.1.6. 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, тестирование

Самостоятельная работа

Дополнительные материалы

Задание на программирование

#### 2.2. Ансамблевые методы и классическое обучение без учителя

##### 2.2.1. Случайный лес и оценка значимости признаков

Лекция

Техника ансамблирования

RSM

Дилемма смещения

Смешивание

Стекинг

Ансамбли деревьев

Оценка значимости признаков

Feature importances

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Задания на программирование

Комплексная задача

##### 2.2.2. 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

Задание на программирование

##### 2.2.3. Кластеризация

Лекция

Введение в кластеризацию

Разнообразие задач кластеризации

K-Means

ЕМ-алгоритм

Агломеративная иерархическая кластеризация  
Графы и методы на основе плотности точек  
Выбор метода кластеризации  
Оценка качества и рекомендации  
Практическая работа  
Выполнение задачи на программирование по теме урока, тестирование  
Самостоятельная работа  
Дополнительные материалы  
Задание на программирование  
2.3. Нейросети и 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  
Задание 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  
Задание 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  
Задания на программирование  
Задание на программирование "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

Задание 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

Задание на программирование

2.3.6. Механизм внимания

Лекция

Механизм внимания Attention. Обзор

Механизм внимания в математической форме

Механизм внимания Self Attention

Механизм внимания Multi-Head Attention

Реализация механизма внимания на Python

Практическая работа

Выполнение задачи на программирование по теме урока

Самостоятельная работа

Дополнительные материалы

Задание на программирование

2.3.7. Архитектура Transformer. BERT в задаче классификации текстов

Лекция

Обзор трансформера

Позиционное кодирование

Нормализация слоев

Преобразователь декодера

ELMo

Обзор модели BERT

Задача языкового моделирования модели BERT

Технические детали работы модели BERT

Практическая работа

Выполнение задачи на программирование по теме урока

Самостоятельная работа

Дополнительные материалы

Задание на программирование

2.3.8. Вопросно-ответные и рекомендательные системы

Лекция

Задача построения вопросно-ответных систем

SQuAD и SberQuAD

Подходы к построению задачи вопросно-ответных систем  
Вопросно-ответная система для открытого контекста  
GPT-2 и GPT-3  
Подробнее о GPT  
Практическая работа  
Выполнение задачи на программирование по теме урока  
Самостоятельная работа  
Дополнительные материалы  
Задание 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. Курс аналитической геометрии и линейной алгебры [Текст], учебник для вузов /Д. В. Беклемишев. -СПб., Лань, 2019
5. Лекции по математическому анализу. В 3 частях, Часть 1, Функции одной переменной, учебник для вузов/Я. М. Дымарский , -Москва, МФТИ, 2020
6. 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

Литература из средств кафедры:

1. Think Python [Электронный ресурс] – Режим доступа - <https://greenteapress.com/wp/think-python-2e/>
2. Automate the Boring Stuff with Python [Электронный ресурс] – Режим доступа - <https://automatetheboringstuff.com/>
3. Dive Into Python 3 [Электронный ресурс] – Режим доступа - <http://diveintopython3.problemsolving.io/>
4. Problem Solving with Algorithms and Data Structures using Python [Электронный ресурс] – Режим доступа - <https://runestone.academy/runestone/static/pythonds/index.html>
5. Swaroop Chitlur. A Byte of Python [Электронный ресурс] – Режим доступа - <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf> – 2020.
6. Марр, Б. Искусственный интеллект на практике: 50 кейсов успешных компаний / Бернард Марр, Мэтт Уорд; пер. с англ. Е. Петровой. – М.: Манн, Иванов и Фербер, 2020. – 316 с.
7. Мин, Ц. Как Alibaba использует искусственный интеллект в бизнесе: Сетевое взаимодействие и анализ данных / Цзэн Мин; пер. с кит. — М.: Альпина Паблишер, 2022. — 360 с.
8. Амейзен Э. Создание приложений машинного обучения: от идеи к продукту. Пер. с англ.— СПб.: Питер, 2023. – 272 с.
9. Bratsis, I. The AI Product Manager's Handbook: Develop a product that takes advantage of machine learning to solve AI problems. – Packt Publishing Ltd, 2023 . – 250 p.
10. Лутц, Марк. Изучаем Python: авторитетный курс объектно-ориентированного программирования / Марк Лутц; перевод с английского Ю. Н. Артеменко. – 5-е изд. – Москва: Диалектика; Санкт-Петербург: Диалектика, 2020.
11. Reitz, K., Schlusser, T. The Hitchhiker's Guide to Python: Best Practices for Development. – O'Reilly, 2016. – URL: [docs.python-guide.org](https://docs.python-guide.org)
12. Kevin Patrick Murphy. Probabilistic Machine Learning: An Introduction. MIT Press, March 2022. URL: <https://probml.github.io/pml-book/book1.html>
13. Элбон К. Машинное обучение с использованием Python. Сборник рецептов: Пер. с англ. – СПб.: БХВ-Петербург, 2020. – 384 с.
14. Дайзенрот М., Фейзал А., Он Ч. Математика в машинном обучении. – СПб.: Питер, 2023. – 512 с.

#### 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. 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.

Дополнительная литература из средств кафедры:

1. Бэрри П. Изучаем программирование на Python [Текст] / П. Бэрри. – 2017.
2. Савельев В. Статистика и котика [Текст] / Владимир Савельев. – 2018.
3. Бослав С. Статистика для всех [Текст] / Сара Бослав. – 2015.
4. Хамидуллин Р. Я. Теория вероятностей и математическая статистика [Текст] / Р.Я. Хамидуллин. – 2020.
5. Справочник по функциям DAX [Электронный ресурс] – Режим доступа - <https://docs.microsoft.com/ru-ru/dax/dax-function-reference>
6. Талер Р. Nudge. Архитектура выбора [Текст] / Ричард Талер. – 2017.
7. Желязны Дж. Говори на языке диаграмм [Текст] : пособие по визуальным коммуникациям / Джин Желязны. – 2020.
8. Федоров Д. Основы программирования на примере языка Python [Текст] : учебное пособие / Д. Федоров. - 2018.
9. Свейгарт Э. Автоматизация рутинных задач с помощью Python [Текст] : практическое руководство для начинающих / Эл Свейгарт. – 2017.
10. Домингос, Педро. Верховный алгоритм: как машинное обучение изменит наш мир / П. Домингос; пер. с англ. В. Горохова. — М. : Манн, Иванов и Фербер, 2016. — 336 с.
11. Кендалл М., Стьюарт Дж. Многомерный статистический анализ и временные ряды. – М.: Наука, 1976. – 736 с.
12. Кокс Д., Хинкли Л. Теоретическая статистика. – М.: Мир, 1978, – 560 с.
13. Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2006

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

1. Документация Postgres про сравнение строк – <https://postgrespro.ru/docs/postgrespro/9.5/functions-matching>
2. Документация Postgres про другие функции работы со строками - <https://postgrespro.ru/docs/postgrespro/9.5/functions-string>
3. Тестер регулярных выражений - <https://www.regextester.com>
4. Интерактивный учебник по SQL - <http://www.sql-tutorial.ru/ru/content.html>
5. Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>
6. Игровой тренажер по Python: [py.checkio.org](http://py.checkio.org)
7. <https://numpy.org/doc/stable/>
8. <https://new.pythonforengineers.com/blog/audio-and-digital-signal-processingdsp-in-python/>
9. <https://www.kaggle.com/code/gabrielmilan/mp3-to-numpy-and-back>
10. <https://github.com/>
11. <https://dataverse.harvard.edu/>
12. [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/index.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html)
13. <https://pbpython.com/pandas-pivot-table-explained.html>
14. <https://proglab.io/p/moem-dataset-rukovodstvo-po-ochistke-dannyh-v-python-2020-03-27>
15. <https://pandas.pydata.org/docs/>
16. [https://pandas.pydata.org/docs/getting\\_started/comparison/index.html](https://pandas.pydata.org/docs/getting_started/comparison/index.html)
17. <https://pbpython.com/groupby-aggr.html>
18. <https://tproger.ru/articles/pandas-data-wrangling-cheatsheet>
19. <https://docs.scipy.org/doc/scipy/reference/stats.html>
20. <https://www.scribbr.com/statistics/confidence-interval/>
21. <https://www.enago.com/academy/evaluate-statistical-hypothesis-testing/>
22. <https://machinelearningmastery.com/statistical-hypothesis-tests-in-python-cheat-sheet/>
23. Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных – <http://www.machinelearning.ru/>
24. Python Numpy Tutorial - <https://cs231n.github.io/python-numpy-tutorial/>
25. Learn Git Branching - [https://learngitbranching.js.org/?locale=ru\\_RU](https://learngitbranching.js.org/?locale=ru_RU)
26. <https://www.deeplearningbook.org/>
27. <https://mml-book.github.io/book/mml-book.pdf>
28. <https://unbiasedresearch.blogspot.com/>
29. <https://habr.com/ru/articles/264241/>
30. [https://scikit-learn.org/stable/supervised\\_learning.html#supervised-learning](https://scikit-learn.org/stable/supervised_learning.html#supervised-learning)
31. Кластеризуем лучше, чем «метод локтя» / Хабр ([habr.com](https://habr.com/ru/companies/jetinfosystems/articles/467745/)) <https://habr.com/ru/companies/jetinfosystems/articles/467745/>
32. Кластеризация - scikit-learn <https://scikit-learn.ru/clustering/>
33. <https://uproger.com/10-bibliotek-python-dlya-mashinnogo-obucheniya/?ref=vc.ru>
34. TensorFlow <https://www.tensorflow.org/?hl=ru>
35. Keras: Deep Learning for humans <https://keras.io/>
36. PyTorch <https://pytorch.org/>
37. Caffe | Deep Learning Framework ([berkeleyvision.org](https://berkeleyvision.org)) <https://caffe.berkeleyvision.org/>
38. <https://towardsdatascience.com/ml-impossible-train-a-1-billion-sample-model-in-20-minutes-with-vae-x-and-scikit-learn-on-your-9e2968e6f385>
39. <https://pub.towardsai.net/python-pandas-vs-vaex-dataframes-a-comparative-analysis-5171636b4ee1>

## 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/>

## **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)

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Программная инженерия передовая инженерная школа радиолокации, радионавигации и программной инженерии центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

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

#### Разработчики:

Р.Г. Нейчев, старший преподаватель  
Г.К. Тарасенко, преподаватель  
Н.А. Долгополов, преподаватель  
Ж.И. Зубцова, канд. физ.-мат. наук, эксперт  
Л.С. Орлова, методист

## 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 цели	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, организовывать обсуждение разных идей и мнений
ОПК-3 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области информатики и вычислительной техники, учитывая особенности и ограничения различных методов решения	ОПК-3.7 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Продвинутые методы машинного обучения» обучающийся должен:

**знать:**

- возможности основных библиотек, используемых для анализа данных;
- как библиотека NumPy помогает в научных вычислениях и обработке данных;
- понятия вектора и матрицы, векторного пространства, нормы вектора, ортогональности и гиперплоскости;
- как выполнять базовые операции над векторами и матрицами;
- 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;
- 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;
- виды, источники и способы хранения данных (csv, tsv-файлы и другие);
- структуры данных и инструменты, предоставляемые библиотекой Pandas для работы с данными;
- терминологию, используемую в машинном обучении;
- виды линейных моделей обучения, метрики измерения качества линейных моделей;
- базовые сведения об ансамблевых моделях;
- производительность ансамблевых моделей;
- в каких задачах машинного обучения используются линейные модели;
- теорему Гаусса-Маркова;
- что такое градиент функции;
- методы оптимизации;
- понятия правдоподобия в задачах машинного обучения;
- как использовать модель логистической регрессии в задачах бинарной и мультиклассовой классификации;
- различные метрики оценки качества классификации;
- когнитивные законы и принципы восприятия информации человеком;
- виды графического представления данных и ситуации их использования;
- методы кластеризации и методы понижения размерности, принципы построения рекомендательных систем;
- метод опорных векторов, используемый для задач классификации и регрессионного анализа;
- способ создания нелинейного классификатора с помощью так называемого ядерного трюка (kernel trick);
- метрики оценки качества классификации;
- ROC-AUC;
- архитектуру Transformer;
- принцип позиционного кодирования;
- как работает декодер в Transformer;
- модель ELMo;
- модель BERT;
- задачу языкового моделирования;
- свойства нормального распределения;
- как использовать библиотеки Python для анализа данных в некоторых задачах машинного обучения;
- терминологию, используемую в нейросетях;
- архитектуры нейронных сетей;
- основные библиотеки Python, используемые для работы с нейросетями;
- возможности библиотеки matplotlib для построения различных видов графиков и настройки их отображения;
- об инструментах для визуализации данных, используемых в работе аналитика;
- функциональные возможности сервиса Yandex DataLens для построения и настройки графиков;
- критерии информативности: энтропию и критерий Джини;
- как использовать решающие деревья в задаче регрессии;
- что такое механизм внимания;
- особенности модели глубокого обучения Seq2Seq, и как ее используют в задаче машинного обучения;
- Self-Attention;
- Multi-Head Attention;
- различные техники ансамблирования и теоретические предпосылки к их применению;

**уметь:**

- использовать базовые операции по работе с массивами, математические и статистические функции библиотеки NumPy для решения прикладных задач;
- решать системы линейных уравнений в Python матричным методом, решать задачи с помощью системы линейных уравнений;
- применять NumPy для работы с векторами и матрицами;
- вычислять косинусную меру близости векторов и использовать ее для нахождения разницы между словами;
- применять метод kNN для решения задач машинного обучения;
- выполнять выгрузку данных с использованием библиотеки Pandas;
- проводить предварительную обработку данных с использованием библиотеки Pandas — получать информацию о DataFrame, работать со строками и столбцами;
- осуществлять группировку и агрегацию таблиц с использованием Pandas;
- строить модели линейной и логистической регрессии с использованием библиотек Python;
- рассчитывать метрику качества линейной и логистической регрессии;
- строить ансамблевые модели — решающее дерево, случайный лес, градиентный бустинг;
- формально поставить задачу линейной регрессии;
- использовать L1- и L2-регуляризации для решения задач машинного обучения;
- решать задачи оптимизации градиентными методами;
- решать задачи линейной классификации в машинном обучении;
- Выполнять практические задачи и проекты в команде;
- 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;
- реализовать методы кластерного анализа на примере искусственных данных, выполнить расчет оценок качества кластеризации с помощью библиотек Python;
- реализовать методы понижения размерности применительно к датасету с помощью Python;
- составить матрицу рейтингов с помощью Python и выполнять операции с ней;
- использовать метод кросс-валидации для оценки качества модели;
- использовать self attention для Transformer;
- кодировать энкодер Transformer;
- использовать метод t-SNE в задаче снижения размерности;
- отличать и понимать базовые статистические концепции — генеральная совокупность, выборка;
- вычислить точечные оценки и доверительные интервалы, интерпретировать их;
- проверять статистические гипотезы с использованием тестов на нормальность данных, равенство дисперсий, сравнение средних, взаимосвязь переменных;
- написать программу для вычисления результата сигмоидальной функции активации для заданных входных данных;
- написать код, реализующий свертку изображения и фильтра;
- реализовать простую нейросеть, состоящую из одного входного слоя, одного скрытого слоя и одного выходного слоя;
- 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;
- строить некоторые виды графиков в Yandex DataLens;
- строить графики и диаграммы с помощью библиотеки Altair, добавлять в графики интерактив и выполнять их настройку;
- строить статистические диаграммы с использованием библиотеки Seaborn;
- использовать решающие деревья в задачах машинного обучения;
- обучать сверточную нейронную сеть (CNN);
- использовать сверточную нейронную сеть для обработки изображений;
- создавать презентации на основе диаграмм с помощью движка Reveal.js, встраивая в него диаграммы из Yandex DataLens;
- создавать интерактивные инфопанели, применяя функционал библиотеки Altair;
- выбрать библиотеки Python для решения задачи анализа данных;
- выполнять анализ данных из одного и нескольких источников с использованием языка Python;
- строить и анализировать матрицу корреляции на Python;

## 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

Примеры заданий на программирование

Задание 1. Базовые операции с массивами NumPy

У вас есть два одномерных массива:

- `array = np.arange(10) ** 4` — массив платежей по играм из сервиса;
- `array_2 = np.arange(10) ** 3` — массив платежей по подпискам в сервисе.

Выполните последовательно следующие задания применительно к исходным массивам:

1. Определите общую сумму дохода сразу по двум потокам. Для этого сначала сложите полученные массивы, после чего получите сумму всех элементов. Результат сохраните в переменную `array_sum`.
  2. Вычислите, насколько больше денег принесла продажа игр за указанный период. Результат сохраните в переменную `array_difference`.
  3. Сохраните 2-й элемент массива по платежам игр в переменную `game_payments2`.
  4. Сохраните последний элемент массива по платежам подписок в переменную `subscription_last`.
- Обратите внимание: в заданиях не требуется выполнять вывод полученных результатов на экран.

Задание 2. Базовые операции с DataFrame

Ваш коллега создал `df` — объект DataFrame и передал в работу вам:

	I	II	III
0	1,0	5,0	NaN
1	2.0	6.0	NaN
2	3.0	NaN	7.0
3	4.0	NaN	6.0

Выполните следующие операции применительно к данному объекту `df`:

1. Замените индексы строк на последовательность чисел от 1 до 4, используя соответствующий метод библиотеки Pandas.
2. Переименуйте названия колонок в последовательность букв A, B, C, используя соответствующий метод библиотеки Pandas.
3. Замените пропущенные значения числом 55.

Обратите внимание: никакие результаты работы программы не требуется выводить на экран.

Задание 3. Расчет статистических показателей подписчиков социальной сети

Вам даны показатели прироста подписчиков за месяц в 300 аккаунтах социальной сети, которые сохранены в объект DataFrame с именем `list_metrics`:

	Прирост
0	-34
1	778
2	888
3	-70
4	322
...	...
295	1
296	261
297	104

298 638  
299 280  
300 rows x 1 columns

Выполните следующие задания применительно к исходным данным:

1. Рассчитайте стандартное отклонение прироста подписчиков. Результат округлите до второго знака после запятой и сохраните в переменную `result1`.
2. Рассчитайте размах прироста подписчиков. Результат округлите до второго знака после запятой и сохраните в переменную `result2`.

Обратите внимание: в заданиях не требуется выполнять вывод полученных результатов на экран.

#### Задание 4. Расчет размера выборки

Определите размер выборки для исследования с доверительным уровнем в 99% и ошибкой не более 1%. При том что мы не знаем, что 80% генеральной совокупности обладают целевым признаком. Результат расчета (целое число) сохраните в переменную `size`.

Обратите внимание: в заданиях не требуется выполнять вывод полученных результатов на экран.

#### Введение в машинное обучение. Метод ближайших соседей

##### Задание 5. Подсчет расстояний для kNN

В рамках этой задачи необходимо заполнить файл `k_nearest_neighbor.py` в папке ([https://github.com/girafe-ai/ml-course/tree/22f\\_basic/homeworks/assignment0\\_01\\_knn](https://github.com/girafe-ai/ml-course/tree/22f_basic/homeworks/assignment0_01_knn)) 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.

#### Методы оптимизации и регрессионного анализа

##### Задание 6. Вычисление производных

В рамках этой задачи необходимо заполнить файл `'derivatives.py'` в папке ([https://github.com/girafe-ai/ml-course/tree/23s\\_dd\\_ml/homeworks/hw03\\_derivatives](https://github.com/girafe-ai/ml-course/tree/23s_dd_ml/homeworks/hw03_derivatives)) и отправить получившийся `.py` файл.

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

##### Задание 7. Ядра для SVM

В рамках этой задачи необходимо заполнить файл `svm.py` по ссылке (<https://contest.yandex.ru/contest/45298/problems/>) и отправить получившийся `.py` файл.

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

##### Задание 8. Распределение Лапласа

В рамках этой задачи необходимо заполнить файл `distribution.py` в папке ([https://github.com/LXDMIPT/ml-course/tree/23s\\_dd\\_ml/homeworks/hw02\\_laplace](https://github.com/LXDMIPT/ml-course/tree/23s_dd_ml/homeworks/hw02_laplace)) 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

Примеры тестовых вопросов

Библиотека NumPy

1. Что показывает атрибут `ndarray.size`?

- Количество элементов массива
- Число измерений массива
- Форму массива
- Типы элементов массива.

2. Что выведет на экран код `print(np.array([1,2,3,4])[-1])`?

- None
- False
- 1
- 4

3. Вам дан следующий фрагмент кода:

```
print(np.arange([1,2,3,4,5,6]).reshape(2, x)).
```

На что необходимо заменить `x`, чтобы программа работала?

- 1
- 2
- 3
- 4

4. Что будет результатом применения функции `transpose()` к одномерному массиву?

- None
- 1
- Ошибка
- Исходный массив

Введение в линейную алгебру для машинного обучения

1. Что является результатом скалярного произведения векторов?

- Скаляр
- Вектор
- Матрица
- Точка в декартовых координатах

2. При каких условиях норма вектора равна нулю?

- Если длина вектора равна нулю
- Если все координаты вектора равны нулю
- Если угол между вектором и осью координат равен нулю
- Если проекция вектора на ось координат равна нулю
- Если сумма координат вектора равна нулю

3. Что такое ортогональная проекция вектора  $u$  на вектор  $v$ ?

- Часть вектора  $u$ , сонаправленная с вектором  $v$
- Скалярное произведение векторов  $u$  и  $v$
- Разность векторов  $u$  и  $v$
- Произведение векторов  $u$  и  $v$
- Часть вектора  $u$ , перпендикулярная вектору  $v$

4. Какой будет размерность транспонированной матрицы размерности  $m \times n$ ?

- $n \times m$
- $m + n$
- $m - n$
- $n - m$

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

наблюдениям

признакам

значениям целевой переменной

6. В задаче классификации целевая переменная ...

принимает только целочисленные значения

является вектором из действительных чисел

принимает значение из конечного множества ответов

7. Выберите правильные утверждения о ROC-кривой.

ROC-кривая построена в осях TPR и FPR

Классификатор с  $ROC\ AUC = 0,55$  способен лучше разделять классы, чем классификатор с  $ROC\ AUC = 0,05$

ROC кривая построена в осях точности (Precision) и полноты (Recall)

ROC AUC может быть оптимизирована напрямую с помощью градиентных методов

ROC кривая определена как для задачи бинарной классификации, так и для задачи регрессии

8. Выберите правильные утверждения о PR-кривой.

PR-кривая построена в осях TPR и FPR

Классификатор с  $PR\ AUC = 1.0$  может обладать ROC AUC меньше 1.0

PR-кривая построена в осях точности (Precision) и полноты (Recall)

PR AUC может быть оптимизирован напрямую с помощью градиентных методов

PR кривая определена как для задачи бинарной классификации, так и для задачи регрессии

9. Решающие деревья обычно строятся с использованием методов...

оптимизации первого порядка

оптимизации второго порядка

жадной оптимизации

случайного поиска

генетического алгоритма

10. В случае наличия пропусков в данных решающие деревья...

требуют предварительного заполнения пропусков

способны автоматически заполнить пропуски

способны сделать предсказание даже при наличии пропусков в данных

не должны применяться

11. Использование SGD с Momentum позволяет...

ускорить сходимость в некоторых случаях.

избежать некоторых локальных минимумов.

избежать седловых точек.

получить аналитическое решение промежуточной оптимизационной задачи.

12. Нейронные сети могут быть представлены в виде комбинации...

решающих деревьев и линейных функций.

линейных функций и нелинейных активаций.

решающих деревьев и нелинейных активаций.

#### 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.