

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Бурятский государственный университет имени Доржи Банзарова»
Институт математики, физики и компьютерных наук
Кафедра информационных систем и методов искусственного интеллекта

Утверждена на заседании
Ученого совета ИМФКН
«___» _____ 20__ г.
Протокол №__

Рабочая программа дисциплины

Технологии программирования

Направление подготовки
02.04.03 Математическое обеспечение и администрирование
информационных систем

Квалификация
Магистр

Форма обучения
Очная

Улан-Удэ

2025

Пояснительная записка

Цели освоения дисциплины

Цель данной дисциплины - освоение основных современных технологий программной реализации алгоритмов машинного обучения.

Место дисциплины в структуре образовательной программы

Дисциплина изучается в 3 семестре и является факультативной направления подготовки 020403 Математическое обеспечение и администрирование информационных систем (магистратура). Базируется на дисциплинах: Программирование, Структуры и алгоритмы компьютерной обработки данных, Машинное обучение, Методы и технологии глубокого обучения.

Планируемые результаты обучения по дисциплине и индикаторы достижения компетенций.

В результате освоения дисциплины студент должен:

Знать:

- принципы, стандартные методы и структуры данных языка Python
- основные приемы и средства подготовки данных с библиотекой Pandas (загрузка, визуализация, очистка, разметка, разбивка, генерация и редукция признаков, стандартизация и другие манипуляции над исходными данными)
- методы и приемы эффективной работы с числовыми массивами с математической библиотекой Numpy
- принципы и методы визуализации данных с библиотекой Matplotlib
- основные принципы и методы работы основных библиотек машинного обучения Scikit-learn, Keras, TensorFlow и PyTorch.

Уметь:

- осуществлять манипуляции по подготовке данных к моделированию в Pandas
- пользоваться основными методами Numpy
- строить и настраивать различные типы графиков с Matplotlib
- строить и оценивать различные модели машинного обучения с помощью специализированных библиотек

Владеть:

- основными технологиями и средствами разработки моделей машинного обучения на языке Python

Планируемые результаты освоения образовательной программы:

ПК-3	Способен руководить проектами по созданию комплексных систем искусственного интеллекта
ПК-3.1	Применяет методы и средства разработки систем искусственного интеллекта

Объем дисциплины в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 1 зачетные единицы, 36 часа.

№ Название разделов дисциплины	Лабораторная работа	Самостоятельная работа
Семестр 3	30	6
1 Введение	2	
2 Библиотеки машинного обучения (1 уровень)	10	3
3 Библиотеки машинного обучения (2 уровень)	18	3

Тематическое планирование курса

Темы

Введение

Семестр 3

Основы языка Python

Лабораторная работа. 2(0) ч. Интерпретатор. Структуры данных. Функции. Работа с ОС. Среды разработки IPython, Spyder, PyCharm.

Библиотеки машинного обучения (1 уровень)

Семестр 3

Numpy

Лабораторная работа. 4(0) ч. Многомерный массив ndarray. Массивы и векторные вычисления. Обработка данных. Линейная алгебра. Генерация случайных чисел.

Самостоятельная работа. 1(0) ч. Многомерный массив ndarray. Массивы и векторные вычисления. Обработка данных. Линейная алгебра. Генерация случайных чисел.

Pandas

Лабораторная работа. 2(0) ч. Структура данных pandas: Series, DataFrame. Базовые функции. Описательная статистика. Обработка пропущенных данных. Индексирование. Допвозможности. Чтение и запись данных, форматы файлов. Переформатирование данных.

Лабораторная работа. 2(0) ч. Многомерный массив ndarray. Массивы и векторные вычисления. Обработка данных. Линейная алгебра. Генерация случайных чисел.

Самостоятельная работа. 1(0) ч. Массивы и векторные вычисления. Обработка данных. Линейная алгебра. Генерация случайных чисел.

Matplotlib

Лабораторная работа. 2(0) ч. API Matplotlib. Графики и подграфики. Настройка элементов графика. Типы графиков: линейные, столбчатые, гистограммы, диаграммы рассеяния и др.

Самостоятельная работа. 1(0) ч. API Matplotlib. Графики и подграфики. Настройка элементов графика. Типы графиков: линейные, столбчатые, гистограммы, диаграммы рассеяния и др.

Библиотеки машинного обучения (2 уровень)

Семестр 3**Scikit-learn**

Лабораторная работа. 4(0) ч. Scikit-learn

Самостоятельная работа. 1(0) ч. Scikit-learn

Keras

Лабораторная работа. 4(0) ч. Keras

Самостоятельная работа. 1(0) ч. Keras

TensorFlow

Лабораторная работа. 4(0) ч. TensorFlow

Самостоятельная работа. 1(0) ч. TensorFlow

PyTorch

Лабораторная работа. 4(0) ч. PyTorch

Заключительная

Лабораторная работа. 2(0) ч. Допбиблиотеки: Statsmodels, Seaborn и т.д. Обзор библиотек Big Data (3 уровень)

БРС

Семестр	Контрольные точки	Баллы
3	Текущий контроль в разделе «Введение»	
	Лабораторная работа	10
3	Текущий контроль в разделе «Библиотеки машинного обучения (1 уровень)»	
	Лабораторная работа	10
	Лабораторная работа	10
	Лабораторная работа	10
3	Текущий контроль в разделе «Библиотеки машинного обучения (2 уровень)»	
	Лабораторная работа	10
	Лабораторная работа	10
3	Зачет	
	Практические задания	40

Итого за семестр 3: 100

Учебно-методическое и информационное обеспечение учебного процесса

Образовательные технологии (в том числе на занятиях, проводимых в интерактивных формах).

Периодически используется технология проблемного обучения. Студентам даются сырые статистические данные из некоторой прикладной области (психология, социология, медицина, экономика и т.д.). Преподаватель формулирует задачу с точки зрения предметной области. Студенту необходимо правильно формализовать задачу и выбрать соответствующий метод анализа, затем решить ее с помощью специализированного программного пакета. Полученный результат необходимо интерпретировать с точки зрения предметной области.

Лекционные и лабораторные занятия проводятся с использованием презентаций.

Учебно-методические материалы, в том числе методические указания для обучающихся по освоению дисциплины

Методические указания к лекционным занятиям.

1. Если во время лекции все же не совсем разобрались в отдельных моментах раскрываемой темы, рекомендуется в индивидуальном порядке уточнить непонятные разделы у преподавателя во время лекции (поднять руку и задать вопрос), либо после нее. Скромность - качество замечательное, но в отдельных случаях быть скромным просто неразумно.
2. Для того, чтобы составлять качественные конспекты лекций, важно понять, что конспект – не дословно записанная речь преподавателя. Преподаватель вообще не обязан диктовать текст лекции под запись, он ведет согласно плану. Таким образом, в течение лекции студент тратит большую часть времени на восприятие информации, меньшую его часть – на ее запись.
3. Для повышения эффективности конспектирования материала рекомендуется воспользоваться следующими рекомендациями: 1) Убирайте только середину слова, а не середину и окончание (например, удачный «эф-ть», не удачный «эф.»). 2) В процессе лекции пишите часть слова, затем в тексте оставляйте место для второй его части, а на перерыве или после занятий (пока не забыли, о чем шла речь) вписывайте оставшуюся часть слова.
- 3) Заменяйте длинные русские слова короткими иностранными, например, несколько – some, выигрывать – win, использовать – use, экономический – economic и т.д.

Методические указания к лабораторным (практическим) занятиям.

1. В ходе лабораторных занятий обучающиеся фактически впервые сталкиваются с самостоятельной практической деятельностью в конкретной области – содействует становлению студентов как будущих специалистов. Поэтому, необходимо студенту проявить здесь особое усердие и получить ощутимый результат.
2. Результаты выполнения лабораторных (практических) работ нужно оформить в виде отчета. Как правило, отчет состоит из 3-х частей: план отчета (общая структура задания); расчетные формулы, блок-схема алгоритма, принципиальная часть программного кода, применяемые методы и средства (библиотеки, модули, структуры данных, службы, шаблоны классов, математические методы ит.п.), авторский проект решения задачи; выводы.
3. Перед сдачей лабораторных работ (практических заданий) необходимо повторить теоретический материал для более глубокого понимания и грамотного комментирования выполненной работы преподавателю.

Методические указания к самостоятельной работе студента.

1. Выполняйте внеаудиторное задание в день его получения, а накануне занятия повторите его.

2. Для успешного выполнения задания создайте условия, которые отвечают требованиям гигиены умственного труда: удобное место, достаточное освещение, тишина, перерывы, необходимое оборудование.
3. Начинайте выполнять задание с его осмысления: определите цель, содержание, степень новизны, уровень усвоения, объем, сроки, этапы и приемы выполнения. Спланируйте и соблюдайте затем последовательность действий. Познакомьтесь с алгоритмом и эталоном выполнения задания.
4. Изучите вначале теоретическую основу задания (закон, правило, первоисточник и др.), затем принимайтесь за практическую работу.
5. Старайтесь выполнять задание самостоятельно, применяя знания и умения, усвоенные ранее.
6. Определите свой оптимальный ритм и режим работы.
7. Помните, что следование рекомендациям научной организации учебного труда экономит время, способствует достижению наилучших результатов.

Оценочные средства

По данной дисциплине разработаны оценочные средства, критерии их оценивания, а также методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций (в приложении).

Список литературы

Перечень основной и дополнительной литературы, необходимой для освоения дисциплины.

Основная

1. [Python и анализ данных](#): научно-популярная литература/У. Маккинли ; [пер. с англ. А. А. Слинкин]. —Москва: ДМК Пресс, 2015. —482 с.
Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=73074
2. [Программирование на языке высокого уровня Python](#): Учебное пособие для вузов/Федоров Д. Ю.. —Москва: Юрайт, 2022. —210 с.
Режим доступа: <https://urait.ru/bcode/492920>
3. [Основы программирования на Python](#): Учебное пособие для вузов/Чернышев С. А.. —Москва: Юрайт, 2021. —286 с.
Режим доступа: <https://urait.ru/bcode/477353>

Дополнительная

1. [Программирование GPU при помощи Python и CUDA](#)/Тоуманнен Б.. —Москва: ДМК Пресс, 2020. —252 с.
Режим доступа: <https://e.lanbook.com/book/179469>
2. [Эволюционные нейросети на языке Python](#)/Омельяненко Я.. —Москва: ДМК Пресс, 2020. —310 с.
Режим доступа: <https://e.lanbook.com/book/179494>
3. [Алгоритмы обучения с подкреплением на Python](#): практическое руководство/Лонца А.. —Москва: ДМК Пресс, 2020. —286 с.
Режим доступа: <https://e.lanbook.com/book/179495>
4. [Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных](#): научно-популярная литература/П. Флах ; пер. с англ.

А. А. Слинкин. —Москва: ДМК Пресс, 2015. —400 с.

Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=69955

**Перечень ресурсов информационно-коммуникационной сети «Интернет»,
необходимых для освоения дисциплины**

Естественный научно-образовательный портал. <http://www.en.edu.ru/>

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

Портал электронного обучения БГУ e.bsu.ru

Система дифференцированного интернет-обучения Hecadem, Moodle.bsu.ru

Личный кабинет преподавателя или студента БГУ <https://my.bsu.ru/>

Электронные библиотечные системы: Руконт, издательство «Лань», Консультант студента

**Описание материально-технической базы, необходимой для осуществления
образовательного процесса по дисциплине**

Лекционная аудитория с проектором. Компьютерный класс для лабораторных занятий.

ПРИЛОЖЕНИЕ

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Бурятский государственный университет имени Доржи Банзарова»
Институт математики, физики и компьютерных наук
Кафедра информационных систем и методов искусственного интеллекта

Фонд оценочных средств по учебной дисциплине

Технологии программирования

Направление подготовки
02.04.03 Математическое обеспечение и администрирование информационных систем

Квалификация
Магистр

Форма обучения
Очная

Улан-Удэ
2025

Паспорт фонда оценочных средств (ФОС) по дисциплине «Технологии программирования»

Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

- ПК-3 Способен руководить проектами по созданию комплексных систем искусственного интеллекта
- ПК-3.1 Применяет методы и средства разработки систем искусственного интеллекта

Этапы формирования компетенции

Семестр	Вид контроля	Оценочные средства
3 семестр	Текущий	Лабораторные работы Выполнение заданий по СРС
	Итоговый (зачет)	Теоретические вопросы

Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Фонд оценочных средств сформирован на основе ключевых принципов оценивания:

- валидность: объекты оценки должны соответствовать поставленным целям обучения;
- определенность: оценочные средства должны быть понятны каждому обучающемуся;
- однозначность: одинаковость оценки качества оценочного средства;
- надежность: использование единообразных показателей и критериев для оценивания достижений.

Описание показателей и критериев оценивания компетенций, а также шкал оценивания

Показатели оценивания компетенций	Уровень сформированности компетенций	Шкала оценивания	Официальный цифровой эквивалент оценки
Знать: - принципы, стандартные методы и структуры данных языка Python - основные приемы и средства подготовки данных с библиотекой Pandas (загрузка, визуализация, очистка, разметка, разбивка, генерация и редукция признаков, стандартизация и другие манипуляции над исходными данными) - методы и приемы эффективной работы с числовыми массивами с математической библиотекой Numpy - принципы и методы визуализации данных с библиотекой Matplotlib - основные принципы и методы работы основных библиотек машинного обучения Scikit-learn,	Высокий	85 – 100 баллов	5 (отлично)
	Базовый	70 – 84 баллов	4 (хорошо)
	Пороговый	60-69 баллов	3 (удовлетворительно)

<p>Keras, TensorFlow и PyTorch.</p> <p>Уметь:</p> <ul style="list-style-type: none"> - осуществлять манипуляции по подготовке данных к моделированию в Pandas - пользоваться основными методами Numpy - строить и настраивать различные типы графиков с Matplotlib - строить и оценивать различные модели машинного обучения с помощью специализированных библиотек <p>Владеть:</p> <ul style="list-style-type: none"> - основными технологиями и средствами разработки моделей машинного обучения на языке Python 			
---	--	--	--

Балльно-рейтинговая система

Для текущего и итогового контроля качества обучения студентов и магистрантов применяется балльно-рейтинговая система, разработанная в соответствии с «Положением об организации учебного процесса с применением кредитно-модульной системы обучения», утвержденным Учебно-методическим советом ФГБОУ ВО «Бурятский государственный университет» от 20 февраля 2012 г. Целью БРС является определение уровня успешности освоения (завершения изучения) обучающимися учебных дисциплин (модулей, циклов) через балльные оценки и рейтинги качества сформированных знаний, умений, профессиональных компетенций, накапливаемые в соответствии с измеряемыми в зачетных единицах трудоемкостями каждого цикла (модуля, дисциплины) и основной образовательной программы в целом.

1. Общая максимальная сумма баллов, которую студент может набрать по дисциплине в течение семестра – 100 баллов: 60 баллов текущий контроль и 40 баллов итоговый контроль (экзамен).

2. Минимальная сумма баллов, при которой студент допускается к экзамену (итоговому контролю), равна 20 баллам.

3. Минимальная сумма баллов, при которой студент получает положительную итоговую оценку по дисциплине равна 60 баллам (60% от 100 баллов).

4. Максимальная оценка за выполнение одной лабораторной работы – 10 баллов.

Связь между четырехбалльной и стобалльной системами оценки качества обучения студентов

Оценка	Рейтинговые баллы
Отлично	80-100
Хорошо	70-80
Удовлетворительно	60-70
Неудовлетворительно	<60

ПРИМЕРЫ ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа №1. Метрический классификатор

Задание. Реализовать метрический классификатор типа k-ближайших соседей – метод парзеновского окна переменной ширины.

Вход: Dataset (размеченная обучающая выборка с категориальными ответами (метки классов))

Выход: k (оптимальное значение); LOO(k) (оценка обобщающей способности); ассигасу (доля правильно классифицированных объектов)

Дополнительные условия.

1. Библиотеки: numpy, pandas, matplotlib, seaborn, sklearn (только для получения датасета) и т.д.

1. Dataset: Iris. Можно выбрать другой датасет из репозитория UCI, но с небольшим количеством количественных признаков.

3. Метрика расстояния: евклидовое.

5. Метрика качества классификации: ассигасу – доля правильно классифицированных объектов.

6. Функция ядра K(r): ядро Епаничникова:

$$K_1(r) = E(r) = \frac{3}{4}(1-r^2)[|r| \leq 1],$$

где [] - нотация Айверсона: [ложь] = 0, [истина] = 1.

7. Скользящий контроль для настройки k: Leave-one-out (LOO) - количество ошибок на контроле.

Требования к реализации.

1. Загрузка стандартного датасета:

```
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data # массив numpy
Y = iris.target # массив numpy
```

2. Модульная структура программы (набор самостоятельных функций). Рекомендации: функция расстояния (Distance), функция ядра (Kernel), определение класса объекта (Predict), скользящий контроль (LOO). Можно реализовать в виде класса NearestNeighbor с соответствующими методами (ООП вариант приветствуется).

3. Построить график LOO(k).

4. Комментарии к коду.

Материалы.

1. Самоучитель Python: <https://pythontutor.ru/>

2. Официальный учебник Python на английском: <https://docs.python.org/3/tutorial/>

- на русском:

https://ru.wikibooks.org/wiki/Python/%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA_Python_3.1

3. Шпаргалки Python-DataScience:

https://www.dropbox.com/sh/gmfsu39jsagyyq9/AADD2w4M3eUF2s1jn_Fk4AMXa?dl=0

4. Мануал по библиотекам Data science: <https://scipy.org/>

5. Сто заданий по Numpy (чем больше сделаете, тем лучше для вас):

<https://github.com/rougier/numpy-100>

6. Лекция Воронцова К.В. «Курс Машинное обучение» 2019:

<https://www.youtube.com/watch?v=SZkrxWhI5qM&list=PLJOzdkh8T5krxc4HsHbB8g8f0hu7973fK> – Машинное обучение. Метрические методы. К.В. Воронцов, Школа анализа данных, Яндекс

Установка Python. Установить сборку Anaconda со всеми необходимыми библиотеками (numpy, pandas, matplotlib, sklearn и т.д. + редакторы кода):

<https://www.anaconda.com/distribution/>

Лабораторная работа №2. Кластерный анализ

Задание. Реализовать алгоритм кластеризации k-means (k-средних, алгоритм Ллойда).

Вход: Dataset (обучающая выборка), k (количество кластеров)

Выход: Метки кластеров для каждого объекта; центры кластеров.

Дополнительные условия.

1. Dataset: Iris. Можно выбрать другой датасет, но с небольшим количеством количественных признаков.
2. Библиотеки: все стандартные библиотеки для Data Science, кроме sklearn.
3. Параметр k (количество кластеров) задается пользователем при запуске.
4. Метрика расстояния: евклидовое
5. Метрика качества кластеризации при фиксированном k: сумма квадратов внутрикластерных расстояний (в каждом кластере рассчитывается сумма квадратов расстояний до центра и суммируются).

Требования к реализации.

1. Датасет в коде должен под переменной X (двумерный массив numpy).
2. Модульная структура программы. Рекомендация: Distance (функция расстояния), InitializationCentre (начальная инициализация центров кластеров), Expectation (кластеризация с текущими центрами), Maximization (перерасчет координат центров), Quality (функционал качества кластеризации). Можно реализовать в виде класса Kmeans (ООП вариант приветствуется).
3. Программа должна быть применима к любому датасету.
4. Так как алгоритм k-means зависит от начальной инициализации центров кластеров нужно в программе несколько раз (например, m=10) получить кластеризацию из различных случайных инициализаций и выбрать среди них наилучшую по метрике качества. Можно количество инициализаций задать вначале программы как параметр.
5. Построить двумерный график (выбрать два признака) с цветными метками реальных классов ирисов и на ней обозначить «крестиками» центры кластеров – нужно для визуальной проверки адекватности кластеризации.
6. Приветствуется (и даже необходимо) применение удобных библиотечных методов и функций.
7. Комментарии в коде.

Материалы.

- Лекция Воронцова К.В. «Курс Машинное обучение» 2019:

<https://www.youtube.com/watch?v=SZkrxWhI5qM&list=PLJOzdkh8T5krxc4HsHbB8g8f0hu7973fK> – Машинное обучение. Кластеризация и частичное обучение. К.В. Воронцов, Школа анализа данных, Яндекс.

- <http://www.machinelearning.ru/wiki/images/5/52/Voron-ML-Clustering-SSL-slides.pdf> – презентация «Кластеризация и частичное обучение», алгоритм Ллойда

Лабораторная работа №3. Метод стохастического градиента для гребневой регрессии

Задание. Реализовать метод стохастического градиента для обучения многомерной линейной регрессионной модели с L2-регуляризацией (гребневая регрессия или ridge-регрессия).

Вход: *Dataset* (размеченная обучающая выборка с вещественными ответами), λ (параметр экспоненциального скользящего среднего), *eps* (параметр останова)

Выход: R^2 (коэффициент детерминации); τ (коэффициент регуляризации), w_i (коэффициенты регрессионной модели)

Дополнительные условия.

1. Библиотеки: *numpy*, *pandas*, *matplotlib*, *seaborn* и т.д.
2. *Dataset*: выбрать из репозитория UCI с небольшим количеством количественных признаков. Исходный датасет разбивается на обучающую (*train*) и тестовую (*test*) выборку. Рекомендация: добавьте в датасет дополнительный фиктивный признак равный 1 на всех объектах, соответствующий пороговому параметру (w_0) модели (для удобства реализации).
3. Функционал качества с регуляризацией (эмпирический риск) :

$$Q = \sum_{i=1}^l (\langle w, x_i \rangle - y_i)^2 + \frac{\tau}{2} \|w\|^2$$

4. Метод обучения: метод наименьших квадратов + стохастический градиентный спуск. Градиентный шаг для регрессии: $w_i = w_{i-1}(1 - h\tau) - (\langle w_{i-1}, x \rangle - y)x$
5. Темп обучения: $h = 1/i$, где i – номер итерации (другие разумные варианты приветствуются).
6. Инициализация весов: $w_i = 0$ (другие разумные варианты приветствуются).
7. Порядок выбора объектов из обучающей выборки: случайный (другие разумные варианты приветствуются).
8. Критерий останова: $Q_i - Q_{i-1} < eps$, где Q_i – сумма квадратов ошибки (невязки) на i -ой итерации, вычисляется (оценивается) по экспоненциальному скользящему среднему. Построить график сходимости Q_i .
9. Контроль переобучения: L2-регуляризация с параметром τ , который настраивается по контрольной выборке (в качестве контрольной выборки брать тестовую). Построить график зависимости качества модели на контрольной выборке от параметра τ . Диапазон и шаг сетки для оптимизации τ определить самостоятельно.
10. Показатель качества модели: коэффициент детерминации R^2 – доля объяснённой изменчивости ответов моделью, вычисляется как квадрат коэффициента корреляции между истинными (y) и предсказанными моделью значениями ответов (\hat{y}). Чем ближе к 1, тем выше качество модели.

Требования и рекомендации к реализации.

1. Рекомендуемые имена переменных: X – матрица объекты-признаки, y – ответы; X_{train} , y_{train} – объекты и ответы обучающей выборки; X_{test} , y_{test} – объекты и ответы тестовой выборки.
2. Модульная структура программы. Рекомендуемые функции: SGD (стохастический градиентный спуск), CrossValidation (скользящий контроль – оценка качества модели на контрольной выборке), Predict (получение ответа от модели). Можно реализовать в виде класса SGD_Ridge с соответствующими методами (ООП вариант приветствуется).
3. Применить возможности *numpy*, в.ч. метод «*dot*» (скалярное произведение), сложение векторов («+») и умножение на число («*») и т.д. – намного упрощает реализацию.
4. Построить график зависимости между истинными и предсказанными ответами на контрольной выборке.
5. Описание датасета в начале программы.

6. Комментарии к коду.

Материалы.

3. Шпаргалки Python-DataScience:

https://www.dropbox.com/sh/gmfsu39jqsagyq9/AADD2w4M3eUF2s1jn_Fk4AMXa?dl=0

4. Мануал по библиотекам Data science: <https://scipy.org/>

5. Сто заданий по Numpy (чем больше сделаете, тем лучше для вас):

<https://github.com/rougier/numpy-100>

6. Лекция Воронцова К.В. «Курс Машинное обучение» 2019:

<https://www.youtube.com/watch?v=SZkrxWhI5qM&list=PLJOzdkh8T5krxc4HsHbB8g8f0hu7973fK> – Машинное обучение. Линейные методы. К.В. Воронцов, Школа анализа данных, Яндекс.

Лабораторная работа №4. Линейные модели классификации: Логистическая регрессия, линейный метод опорных векторов (Linear SVM)

Задание. Сравнить качество обучения двух линейных моделей классификации (логистическая регрессия и Linear SVM) на выбранном датасете с помощью библиотеки sklearn (модули LogisticRegression и LinearSVC).

Вход: Dataset (размеченная обучающая выборка с метками классов)

Выход: accuracy и C для LogisticRegression; accuracy и C для LinearSVC.

Дополнительные условия.

1. Датасет. Самостоятельно выбрать понятную двух-классовую задачу из репозитория UCI. Разбить выборку на обучающую и тестовую. Объем выборки: не слишком большой и не слишком маленький. Количество признаков: не больше 30. Тип признаков: все количественные.
2. Библиотеки: все стандартные для Data Science + mglearn (из книги Мюллера и Гвидо)
3. Метрика качества модели: accuracy (доля правильно классифицированных объектов)
4. Контроль переобучения: L2-регуляризация для обеих моделей. Настроить параметр регуляризации C по тестовой выборке. Построить график зависимости качества модели на тестовой выборке от параметра C.
5. Коэффициенты модели: построить график коэффициентов модели при различных значениях параметра регуляризации C (при 3-4 значениях, среди которых одно оптимальное).
6. Визуализация: построить двумерный график границы принятия решений по двум признакам с соответствующими максимальными значениями коэффициентов.
7. Выводы сравнительного анализа двух линейных моделей на выбранном датасете.

Требования к реализации.

1. В начале программы привести краткое описание датасета (формулировка задачи, описание признаков, целевая переменная).
2. Комментарии к коду. Комментарии к выводу результата.

Материалы.

- логистическая регрессия: <https://www.youtube.com/watch?v=--gJR8pd-jg&list=PLJOzdkh8T5krxc4HsHbB8g8f0hu7973fK&index=6> (Курс «Машинное обучение»

2019)

- метод опорных векторов (SVM линейный):

<https://www.youtube.com/watch?v=jA9CpUSaSN4&list=PLJOzdkh8T5krxc4HsHbB8g8f0hu7973fK&index=4>

- практический путеводитель: книга Мюллера и Гвидо (раздел «Линейные модели», подраздел «Линейные модели для задач классификации»):

<https://www.dropbox.com/sh/v7igfuqugyi54se/AABxePlmv4LWGqnbk97-RTZoa?dl=0> (папка «На русском»)

Лабораторная работа №5. «Наивный Байес»

Задание. Построить два вида наивного байесовского классификатора с параметрической оценкой плотности распределения:

- 1) с предположением о нормальном распределении выборок;
- 2) с предположением о лапласовском распределении выборок.

Вход: датасет для классификации с числовыми признаками

Выход: оценка качества классификации для каждого алгоритма на контрольной и тестовой выборке

Дополнительные условия.

Построить графики одномерных эмпирических и теоретических (нормального, лапласовского) распределений по каждому признаку для одного (любого) класса.

Провести сравнительный анализ двух алгоритмов.

Требования к реализации.

1. Python версии 3.7 и выше
2. Библиотеки: pandas, numpy, matplotlib и др. кроме Sklearn
4. Использовать следующие обозначения переменных: M – матрица матожиданий, D (или Σ) – матрица дисперсий
5. Комментарии к коду.
6. Источник для выбора датасета (можно другие): <https://archive.ics.uci.edu/ml/index.php>

Литература.

1. [Машинное обучение \(курс лекций, К.В.Воронцов\)](#)
2. [Машинное обучение. Вводная лекция. К.В. Воронцов, Школа анализа данных, Яндекс. - YouTube](#) – 11 лекция

Контрольные вопросы.

1. Как выглядит формула Байеса и откуда она вытекает?
2. Какие подзадачи необходимо решить для построения классификатора?
3. При каких допущениях байесовский классификатор становится «наивным»?
4. Какова идея параметрических методов оценки плотности распределения?
5. В чем идея непараметрических методов оценки плотности распределения?
6. Какова идея оценки плотности распределения смесью параметрических распределений?

Лабораторная работа №6. TensorFlow: реализация эквиваленции

Задание. Реализовать операцию логической равнозначности (эквиваленция, XNOR) с помощью нейронной сети на TensorFlow в трех вариантах:

- с функцией активации типа «сигмоида»;
- с линейной функцией активации.
- с функцией активации типа «ReLU»;

Провести сравнительный анализ результатов.

Вход: Таблица истинности для эквиваленции

Выход: Три варианта обученных нейронных сетей с ответами на обучающей выборке

Дополнительные условия.

Построить вычислительный граф для каждой сети средствами TensorFlow и уметь её интерпретировать.

Требования к реализации.

1. Python версии 3.7.6

2. Библиотеки: TensorFlow 2.x, numpy, matplotlib
3. Допустимы два варианта реализации на TF версии 2.x:
 - а) реализация под TF версии 1.x (устарело), совмещенная с помощью модуля tf.compat.v1;
 - б) прямая реализация на TF версии 2.x (рекомендуется).
4. Использовать стандартные обозначения переменных: w – веса, b – пороговый коэффициент и т.д. (согласно основной литературе – по Сантану Паттанаяк)
5. Комментарии к коду.

Литература.

1. Глубокое обучение и TensorFlow для профессионалов. Математический подход к построению систем искусственного интеллекта на Python», Сантану Паттанаяк, 480 стр., ISBN 978-5-907144-25-5, «ДИАЛЕКТИКА», 2019.

Ссылка для скачивания (сканы основных страниц, название файла:

Scan_DeepLearning_TensorFlow_СантануПаттанаяк):

<https://www.dropbox.com/sh/v7igfuqugyi54se/AABxePlmv4LWGqnbk97-RTZoa?dl=0>

Ссылка для приобретения: <https://www.ozon.ru/product/glubokoe-obuchenie-i-tensorflow-dlya-professionalov-matematicheskij-podhod-k-postroeniyu-sistem-151531710/?from=bar&partner=wsht&sh=sJglGPYx>

2. Tensorflow 2.x - уроки (для Python):

<https://www.youtube.com/playlist?list=PLA0M1Bcd0w8ynD1umfubKq1OBYRXhXkmH>

3. Официальный туториал по TensorFlow: <https://www.tensorflow.org/tutorials>

Контрольные вопросы.

1. Какова архитектура сети (нарисовать)?
2. Как выглядят линейная функция активации, «сигмоида» и ReLU?
3. Как задаются начальные значения весов сети, из какого распределения?
4. Какого вида функция потерь используется?
5. Зависит ли сходимость обучения от выбора метода оптимизации?

Критерий оценивания лабораторных работ

Форма сдачи лабораторной работы состоит из двух частей: практическая часть и устные комментарии и объяснения. Оценивается работа по следующим принципам:

1. Оценивается лабораторная работа если практическая часть выполнена верно (небольшие не точности допускаются).

2. Если студент практическую часть выполнил правильно и смог подробно ее объяснить, то получает соответствующий максимальный балл, указанный в заголовке лабораторной работы.

3. Отнимается «1» балл за некорректные объяснения и комментарии.

4. Отнимается «2» балл за грубые ошибки в объяснениях и комментариях.

5. Оценивается работа в «0» баллов при отсутствии объяснений и комментариев по существу.

Студент в праве пересдать лабораторную работу, если получил балл ниже максимального.

Максимальный суммарный балл за лабораторные работы составляет 50 баллов.

СПИСОК ВОПРОСОВ НА ЗАЧЕТ

I. Основы программирования на Python

1. Перечислите основные структуры данных в Python и приведите примеры их использования.
2. Чем отличаются списки (list) и кортежи (tuple)? Какие преимущества и недостатки каждого типа?
3. Как работают словари (dict) в Python? Опишите операции добавления, изменения и удаления элементов.
4. Напишите простой цикл for, выполняющий итерации по списку чисел и выводящий квадрат каждого числа.
5. Объясните принцип работы условных операторов if, elif, else.
6. В чём разница между функциями range() и enumerate()?
7. Какие встроенные функции Python полезны для работы с массивами данных?
8. Различия между локальными и глобальными переменными.
9. Как реализовать рекурсивную функцию на примере факториала числа?
10. Что такое замыкания в Python и как они реализуются?

II. Библиотеки Python для анализа данных и машинного обучения

11. Какие библиотеки Python часто используются для научных расчетов и анализа данных?
12. Для чего предназначена библиотека NumPy и её главные возможности?
13. Что представляет собой объект массива в NumPy (ndarray), как создать и манипулировать им?
14. Объясните структуру DataFrame в библиотеке Pandas и почему она полезна в анализе данных.
15. Приведите примеры операций фильтрации, сортировки и группировки данных в Pandas.
16. Какое предназначение библиотеки Matplotlib и Seaborn? Покажите различия в применении обеих библиотек.
17. Для чего служит библиотека Scikit-Learn и перечислите её базовые модули.
18. Назовите основные шаги построения модели машинного обучения с использованием Scikit-Learn.
19. Какие функции позволяют оценить качество моделей в Scikit-Learn?
20. Кратко охарактеризуйте библиотеку TensorFlow и её отличия от PyTorch.

III. Реализация основных алгоритмов машинного обучения на Python

21. Опишите реализацию линейной регрессии с использованием библиотеки Scikit-Learn.
22. Что означает классификация методом ближайших соседей (KNN) и как её настроить в Python?
23. Создайте простейшую модель дерева решений и оцените её эффективность.
24. Чем различаются методы обучения Random Forest и Gradient Boosting Tree?
25. Как реализуется логистическая регрессия в Python и какую роль играет сигмоидная функция?
26. Проиллюстрируйте процесс разделения набора данных на тренировочный и тестовый наборы с помощью библиотеки Scikit-Learn.
27. Реализуйте простую модель кластеризации методом K-means и поясните выбор количества кластеров.
28. Постройте модель глубокого обучения с применением библиотеки Keras/TensorFlow.
29. Что такое Pipeline в Scikit-Learn и какова его роль в процессе машинного обучения?
30. Покажите, как выполняется нормализация данных с использованием MinMaxScaler и StandardScaler.

IV. Анализ и подготовка данных

31. Почему важна предварительная обработка данных в машинном обучении?
32. Какие способы очистки данных существуют и как их применять в Python?
33. Приведите примеры методов нормализации данных в Python.
34. Что такое кодирование категориальных признаков (One-Hot encoding, Label encoding) и как оно делается в Python?
35. Опишите методы борьбы с пропущенными значениями.
36. Что такое мультиколлинеарность и как её выявить и устранить средствами Python?
37. Покажите процесс визуализации данных с помощью Matplotlib и Seaborn.
38. Как применить корреляционный анализ в Pandas?
39. Приведите методы проверки распределения данных и выбора подходящей трансформации.
40. Каково значение эксплорационного анализа данных (EDA) и каким образом он осуществляется в Python?

Критерии оценки ответа на теоретический вопрос

Критерий	Требования	Баллы
Знание и понимание теоретического материала	Рассматриваемые понятия определяются четко и полно, приводятся соответствующие примеры	10
	Используемые понятия строго соответствуют теме	8
Анализ и оценка информации	Объясняются альтернативные взгляды на рассматриваемую проблему	8
	Дается личная оценка проблеме	6
Построение суждений	Изложение ясное и четкое	4
	Приводятся различные точки зрения и их личная оценка (примеры применения, плюсы и минусы)	4
Итого		40