

AI System

Курс состоит из двух блоков:

- Базы знаний и онтологии
- Классические методы МО

Каждый блок включает в себя несколько лабораторных работ, для сдачи необходимо сделать R&D отчет по итогам модуля. Защита лабораторных работ происходит в таком формате:

- Показываете ЛР в виде рабочего проекта - если всё ок, значит можно приступить к следующему заданию. Показывать две за раз можно, но если есть замечания, то на следующей паре можно только показывать ЛР, а не защищать модуль.
- Для закрытия модуля и получения оценки необходимо рассказать про блок выполненных заданий с точки зрения постановки задачи, требований, ограничений, назначения.

Модуль 1. Базы знаний и онтологии

Лабораторная 1.

Часть 1. Создание базы знаний и выполнение запросов в Prolog

Часть 2. **Создание онтологии в Protege**

Лабораторная 2. **Разработка системы поддержки принятия решения на основе базы знаний или онтологии**

Модуль 2. Методы МО

Лабораторная 3. Линейная регрессия

Лабораторная 4. **Метод k-ближайших соседей**

Лабораторная 5. Деревья решений

Лабораторная 6. Логистическая регрессия

Модуль 1. Базы знаний и онтологии

Лабораторная 1.

Часть 1. Создание базы знаний и выполнение запросов в Prolog

Требуется создать базу знаний в языке программирования Prolog и реализовать набор запросов, используя эту базу знаний. Задача направлена на развитие навыков работы с фактами, предикатами, и правилами в логическом программировании.

Задание

- **Создание базы знаний:**

Создайте базу знаний. База знаний должна включать в себя **не менее 20 фактов с одним аргументом, 10-15 фактов с двумя аргументам, которые дополняют и показывают связь с другими фактами и 5-7 правил**. Факты могут описывать объекты, их свойства и отношения между ними. Факты 2 и более аргументами могут описывать различные атрибуты объектов, а правила - логические законы и выводы, которые можно сделать на основе фактов и предикатов.

- **Выполнение запросов:**

Напишите несколько запросов для БЗ. Запросы **должны быть разной сложности** и включать в себя:

- Простые запросы к базе знаний для поиска фактов.
- Запросы, использующие логические операторы (**и, или, не**) для формулирования сложных условий (или использовать логические операторы в правилах).
- Запросы, использующие переменные для поиска объектов с определенными характеристиками.
- Запросы, которые требуют выполнения правил для получения результата.

- **Документация:**

В коде должны быть комментарии описания фактов, предикатов и правил.

Критерии оценки

- Корректность базы знаний и выполненных запросов.
- Сложность и разнообразие запросов.
- Качество документации и комментариев к коду.

Тематики БЗ

Любая, связанная с играми. Например,

- Видеоигры
- Правила настольных игр
- Профили игроков
- Игровые персонажи
- История игры
- Игровые механики

Часть 2. Создание онтологии в Protege

Целью этой лабораторной работы является знакомство со средой разработки онтологий Protege и перевод базы знаний, созданной в предыдущей лабораторной работе в онтологическую форму в Protege.

Задание

Преобразовать факты и отношения из Prolog в концепты и свойства в онтологии. Описать классы и свойства в онтологии, которые соответствуют объектам и отношениям из базы знаний. Например, если у были классы "Человек" и "Машина" и свойство "возраст", создайте аналогичные классы и свойства в онтологии в Protege.

Критерии оценки

- Корректное создание онтологии в Protege на основе базы знаний в Prolog.
- Качество перевода фактов, предикатов и отношений из Prolog в онтологию.
- Определение классов, свойств и иерархии классов в Protege.
- Тестирование онтологии и демонстрация ее функциональности (визуализация и проверка запросов).

Лабораторная 2. Разработка системы поддержки принятия решения на основе базы знаний или онтологии

Целью этой лабораторной работы является разработка программы (рекомендательной системы), которая будет использовать базу знаний или онтологию для предоставления рекомендаций на основе введенных пользователем данных. (Knowledge-based support system)

Задание

- Создать программу, которая позволяет пользователю ввести запрос через командную строку. Например, информацию о себе, своих интересах и предпочтениях в контексте выбора видеоигры - на основе фактов из БЗ (из первой лабы)/Онтологии(из второй).
- Использовать введенные пользователем данные, чтобы выполнить логические запросы к БЗ/Онтологии.
- На основе полученных результатов выполнения запросов, система должна предоставить рекомендации или советы, связанные с выбором из БЗ или онтологии.
- Система должна выдавать рекомендации после небольшого диалога с пользователем.

Пример

Входная строка:

Мне 13 лет, мне нравятся: RPG, инди-игры

Нужно

Спарсить строку, разбить на факты, построить запрос, используя эти предикаты. (Формат входной строки фиксированный, искать частичное соответствие подстроки не нужно)

Критерии оценки

- Корректность и эффективность реализации системы поддержки принятия решения.
- Способность программы адекватно использовать базу знаний или онтологию для выдачи рекомендаций.
- Качество тестирования и обработка ввода пользователя.
- Качество документации и описание работы системы.

Модуль 2. Методы МО

В модуле 4 лабораторные работы. Реализовывать можно на любом языке. По каждой лабораторной необходимо сделать небольшой отчёт с результатами выполнения работы. По завершению модуля - большой отчёт.

Лабораторная 3. Линейная регрессия

Линейная регрессия. Теоретическая часть (1).

Задание

- Выбор датасетов:
 - Студенты с **четным** порядковым номером в группе должны использовать набор данных о жилье в Калифорнии [Скачать тут](#)
 - Студенты с **нечетным** порядковым номером в группе должны использовать про обучение студентов
- Получите и визуализируйте (графически) статистику по датасету (включая количество, среднее значение, стандартное отклонение, минимум, максимум и различные квантили).
- Проведите предварительную обработку данных, включая обработку отсутствующих значений, кодирование категориальных признаков и нормировка.
- Разделите данные на обучающий и тестовый наборы данных.
- Реализуйте линейную регрессию с использованием метода наименьших квадратов без использования сторонних библиотек, кроме NumPy и Pandas (для использования коэффициентов использовать библиотеки тоже нельзя). Использовать минимизацию суммы квадратов разностей между фактическими и предсказанными значениями для нахождения оптимальных коэффициентов.

- Постройте **три модели** с различными наборами признаков.
- Для каждой модели проведите оценку производительности, используя метрику коэффициент детерминации, чтобы измерить, насколько хорошо модель соответствует данным.
- Сравните результаты трех моделей и сделайте выводы о том, какие признаки работают лучше всего для каждой модели.
- Бонусное задание
 - Ввести синтетический признак при построении модели

Лабораторная 4. Метод k-ближайших соседей

Задание

Выбор датасета:

Четный номер в группе - Датасет о вине

Нечетный номер в группе - Датасет про диабет

- Проведите предварительную обработку данных, включая обработку отсутствующих значений, кодирование категориальных признаков и масштабирование.
- Получите и визуализируйте (графически) статистику по датасету (включая количество, среднее значение, стандартное отклонение, минимум, максимум и различные квантили), постройте 3d-визуализацию признаков.
- Реализуйте метод k-ближайших соседей без использования сторонних библиотек, кроме NumPy и Pandas.
- Постройте две модели k-NN с различными наборами признаков:
 - Модель 1: Признаки случайно отбираются .
 - Модель 2: Фиксированный набор признаков, который выбирается заранее.
- Для каждой модели проведите оценку на тестовом наборе данных при разных значениях k. Выберите несколько различных значений k, например, k=3, k=5, k=10, и т. д. Постройте матрицу ошибок.

Лабораторная 5. Деревья решений

Деревья решений. Теоретическая часть (1)

Задание

1. Для студентов с четным порядковым номером в группе – датасет с классификацией грибов, а нечетным – датасет с данными про оценки студентов инженерного и педагогического факультетов (для данного датасета нужно ввести метрику: студент успешный/неуспешный на основании грейда)
2. Отобрать **случайным** образом \sqrt{n} признаков
3. Реализовать без использования сторонних библиотек построение дерева решений (дерево не бинарное, numpy и pandas использовать можно, использовать список списков для реализации дерева - нельзя) для решения задачи бинарной классификации
4. Провести оценку реализованного алгоритма с использованием Accuracy, precision и recall
5. Построить кривые AUC-ROC и AUC-PR (в пунктах 4 и 5 использовать библиотеки нельзя)

Лабораторная 6. Логистическая регрессия

логистическая регрессия.docx

1. Выбор датасета:
 - Датасет о пассажирах Титаника: Titanic Dataset
 - Датасет о диабете: Diabetes Dataset
2. Загрузите выбранный датасет и выполните предварительную обработку данных.

3. Получите и визуализируйте (графически) статистику по датасету (включая количество, среднее значение, стандартное отклонение, минимум, максимум и различные квантили).
4. Разделите данные на обучающий и тестовый наборы в соотношении, которое вы считаете подходящим.
5. Реализуйте логистическую регрессию "с нуля" без использования сторонних библиотек, кроме NumPy и Pandas. Ваша реализация логистической регрессии должна включать в себя:
 - Функцию для вычисления гипотезы (sigmoid function).
 - Функцию для вычисления функции потерь (log loss).
 - Метод обучения, который включает в себя градиентный спуск.
 - Возможность варьировать гиперпараметры, такие как коэффициент обучения (learning rate) и количество итераций.
6. Исследование гиперпараметров:
 - Проведите исследование влияния гиперпараметров на производительность модели. Варьируйте следующие гиперпараметры:
 - Коэффициент обучения (learning rate).
 - Количество итераций обучения.
 - Метод оптимизации (например, градиентный спуск или оптимизация Ньютона).
7. Оценка модели:
 - Для каждой комбинации гиперпараметров оцените производительность модели на тестовом наборе данных, используя метрики, такие как accuracy, precision, recall и F1-Score.

Сделайте выводы о том, какие значения гиперпараметров наилучшим образом работают для данного набора данных и задачи классификации. Обратите внимание на изменение производительности модели при варьировании гиперпараметров.

Модуль 2. Шаблон отчёта