Лабораторная работа №2
«Коллизии. Классические методы»

Задание №1
«Формирование датасета»

1. Создать генератор данных, зависящий от кол-ва признаков и объема выборки. Данные должны иметь следующий вид:

Состояние объекта №1			Состояние объекта №2			Коллизия
Признак 1	…	Признак n	Признак 1	…	Признак m	Да / нет

   Под состоянием агента/объекта могут пониматься множество таких признаков как «местоположение», «направление», «угол поворота», «форма объекта» и т. п. и т. д. Необходимо, чтобы в каждом состоянии были представлены 4 вида признаков по возможности (если признаков больше 7):

   - *бинарный*.
   - *номинальный* / конечное неупорядоченное множество.
   - *порядковый* признак / конечное упорядоченное множество.
   - *количественный* признак / бесконечное множество.
2. Сгенерировать 12 датасетов, которые имеют следующие параметры:
   По объему:
   - От 30 по 100;
   - От 100 по 500;
   - От 500 по 1000;
   - От 1000 и выше.

   По количеству признаков:
   - 4–7;
   - 8–10;
   - 10 и больше.

Задание №2
«Выбор алгоритма»

1. Выбрать 4 классических метода машинного обучения и обосновать выбор (в комментариях).
2. Выбрать 3 наиболее быстрых алгоритма, с наименьшим потреблением данных.
3. Сохранение модели.

Лабораторная работа №3
«Оптимизация гиперпараметров»

Генетические алгоритмы

Задание №1
«Классический генетический алгоритм»

1. Используя библиотеку PyGAD или любую другую реализацию (можно собственную), произвести тонкую настройку гиперпараметров для 2 лучших моделей из предыдущей лабороторной.
2. Оценить полученные модели.
3. Сохранить модели.

Задание №2
«Алгоритм роя частиц»

1. Используя библиотеку PySwarms или любую другую реализацию (можно собственную), произвести тонкую настройку гиперпараметров для 2 лучших моделей из предыдущей лабороторной.
2. Оценить полученные модели.
3. Сохранить модели.

Задание №3
«Алгоритм *NEAT*»

1. Используя самый маленький набор данных по объему и количеству признаков и самый большой по тем же критериям, с помощью алгоритма NEAT найти оптимальную нейронную сеть под каждый из наборов.
2. Сравнить модели.
3. Сохранить модели.

Задание №4
«Алгоритм *ES-HyperNEAT*»

1. Используя самый маленький набор данных по объему и количеству признаков и самый большой по тем же критериям, с помощью алгоритма ES-HyperNEAT найти оптимальную нейронную сеть под каждый из наборов.
2. Сравнить модели.
3. Сохранить модели.

Лабораторная работа №4
«Обучение с подкреплением»

Задание №1
«Реализация среды»

1. Используйте библиотеку для создания игровой среды, например, *PyGame* или *gym* (с кастомной средой).
2. Создайте простую двумерную карту с препятствиями (стены, блоки) и целью.
3. Танк и цель должны быть представлены как объекты с координатами и углом поворота.

Задание №2
«Определение системы»

1. Определение состояния, действий и награды (пример):
    a. Состояние (State):
        i. Положение танка (x, y).
        ii. Угол поворота танка.
        iii. Расстояние до цели.
        iv. Угол до цели относительно танка.
        v. Наличие препятствий вблизи (например, расстояние до ближайшего препятствия в четырёх направлениях).
    b. Действия (Action):
        i. Движение вперёд.
        ii. Движение назад.
        iii. Поворот влево.
        iv. Поворот вправо.
        v. Стрельба.
    c. Награда (Reward):
        i. +100 за попадание в цель.
        ii. -10 за столкновение с препятствием.
        iii. -1 за каждый шаг (штраф за время).
        iv. +10 за приближение к цели.
        v. -5 за удаление от цели.

Задание №3
«Реализация агента»

1. Используйте алгоритм обучения с подкреплением, например, Q-learning, Deep Q-Network (DQN) или Policy Gradient.
2. Реализуйте функцию выбора действия на основе текущей политики (например, ε-жадная стратегия для Q-learning).
3. Обучите агента в среде, сохраняя результаты обучения (награды за эпизоды).

Задание №4
«Тестирование и анализ»

1. Протестируйте обученного агента в среде.
2. Визуализируйте процесс обучения (график накопленной награды за эпизоды).

3. Проверьте, как агент справляется с изменением начальных условий (разные стартовые позиции танка и цели).

Лабораторная работа №5 «Интеграция»

Python embedding in C#

Задание №1 «Подготовка»

1. Реализовать 6 скриптов с полной предобработкой сырых данных и прогонку через модели из предыдущей лабораторной (Pipeline).

Задание №2 «Стратегии коллизий»

2. Реализовать на C# стратегии коллизий (подключить 6 скриптов) с помощью Python.NET.
3. Сформировать новые данные.
4. Произвести анализ быстродействия и качества данных стратегий.