

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ  
Методические указания к лабораторным работам  
(второй семестр)

Составитель: С. М. Наместников

Ульяновск

2022

УДК 621.394.343 (076)

ББК 32.88 я7

ПЗЗ

Рецензент: Deep Learning Engineer компании Huawei, канд. техн. наук, Смирнов П.В.

Одобрено секцией методических пособий научно-методического совета Университета

Методы машинного обучения: методические указания к лабораторным работам (второй семестр) /сост. С. М. Наместников. – Ульяновск : УлГТУ, 2022. – 13 с.

Методические указания по курсу «Методы машинного обучения» для студентов направления 11.04.02 Инфокоммуникационные технологии и системы связи, профиль подготовки " Искусственный интеллект и анализ больших данных в обработке изображений " разработаны в соответствии с программой курса «Методы машинного обучения». Лабораторные работы посвящены исследованию и разработки основных методов машинного обучения с использованием языка Python.

Сборник подготовлен на кафедре «Телекоммуникации».

УДК 621.394.343 (076)

ББК 32.88 я7

© С. М. Наместников, составление, 2022

## **СОДЕРЖАНИЕ**

### **Лабораторная работа №1**

Исследование работы алгоритма Ллойда для задачи кластеризации данных

### **Лабораторная работа №2**

Использование метрических регрессионных методов для решения задачи аппроксимации функциональной зависимости

### **Лабораторная работа №3**

Исследование работы алгоритма DBSCAN для задачи кластеризации данных

### **Лабораторная работа №4**

Реализация решающих деревьев для задач классификации

### **Лабораторная работа №5**

Применение случайного леса (random forest) в задачах регрессии

## **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

## **Лабораторная работа №1**

### **Исследование работы алгоритма Ллойда для задачи кластеризации данных**

**Цель работы:** реализовать и провести анализ работы алгоритма Ллойда для решения задачи кластеризации заданной выборки.

#### **Теоретический материал**

Теория для выполнения лабораторной работы доступна на следующих страницах сайта:

<https://proproprogs.ru/ml>

в разделах:

- Задачи кластеризации. Постановка задачи
- Алгоритм кластеризации Ллойда (K-средних, K-means)

А также в соответствующих видеоматериалах, размещенных на странице сайта:

<http://tk.ulstu.ru/video.php?id=3>

#### **Задания на лабораторную работу (по вариантам)**

1. Необходимо построить (реализовать на языке Python) алгоритм Ллойда для кластеризации, следующих данных выборки (для своего варианта):

<http://tk.ulstu.ru/files/clusters.py>

2. Для оценки расстояний между отсчетами использовать метрику Минковского:

$$\rho(x, x_i) = \left( \sum_{j=1}^n \omega_j \cdot |x^j - x_i^j|^p \right)^{1/p}$$

с параметром  $p = 1$  и  $p = 2$  (получится два алгоритма).

3. Число кластеров  $K$  взять равным 2, 3 и 4.

4. Отобразить результаты кластеризации в виде графика точек на плоскости (объекты разных классов должны быть иметь разные маркеры и цвет).

#### **Содержание отчета**

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Текст программы алгоритма Ллойда на языке Python.
3. Графики полученных результатов кластеризации.
4. Выводы по полученным результатам.

## Лабораторная работа №2

### Использование метрических регрессионных методов для решения задачи аппроксимации функциональной зависимости

**Цель работы:** научиться использовать формулу Надарая-Ватсона для решения задач восстановления функциональной зависимости по множеству объектов обучающей выборки.

#### Теоретический материал

Теория для выполнения лабораторной работы доступна на странице сайта:

<https://proproprogs.ru/ml>

в разделах:

- Метрические методы классификации. Метод k ближайших соседей
- Методы парзеновского окна и потенциальных функций
- Метрические регрессионные методы. Формула Надарая-Ватсона

а также в соответствующих видеоматериалах, размещенных на странице сайта:

<http://tk.ulstu.ru/video.php?id=3>

#### Задания на лабораторную работу (по вариантам)

1. Для заданной вариантом функциональной зависимости, искаженной гауссовским шумом  $\{\varepsilon_i\}$  с нулевым средним и единичной дисперсией, выполнить восстановление функционального закона (природы зависимости в данных) с использованием формулы Надарая-Ватсона.

2. В качестве метрики расстояния до отсчетов используйте функцию модуля:

$$\rho(x, x_i) = |x - x_i|, \quad i = 1, 2, \dots$$

3. Исследуйте работу алгоритма со следующими функциями ядра:

- $K(r) = \exp\left(-\frac{r^2}{2}\right)$  - гауссовское;
- $K(r) = |1 - r| \cdot [ |r| \leq 1 ]$  - треугольное;
- $K(r) = [ |r| \leq 1 ]$  - прямоугольное.

и разной шириной окна  $h$  (выберите самостоятельно несколько разумных значений, чтобы увидеть эффект от применения этого параметра).

4. Выведите полученные результаты в виде серии графиков, причем в каждом графике (координатных осях) должно быть по два графика: исходный и восстановленный.

Вариант	Функция
1	$y(x) = \frac{\cos(20x)}{x+0,1} + 0,5 \cdot \varepsilon, \quad x \in [0; 4; 0,01]$
2	$y(x) = \sin(5x) \cdot \cos(x) + 0,1 \cdot \varepsilon, \quad x \in [0; 3; 0,01]$
3	$y(x) = \ln(x) \cdot e^{-x} + 0,01 \cdot \varepsilon, \quad x \in [1; 10; 0,01]$
4	$y(x) = \sin(x) \cdot x^2 + \varepsilon, \quad x \in [0; 3; 0,01]$
5	$y(x) = \sin(5x) \cdot e^{-x} + 0,1 \cdot \varepsilon, \quad x \in [0; 10; 0,01]$
6	$y(x) = \frac{5}{10+x^2} + 0,05 \cdot \varepsilon, \quad x \in [0; 20; 0,1]$
7	$y(x) = \cos(5x) \cdot e^x + 5 \cdot \varepsilon, \quad x \in [0; 4; 0,01]$
8	$y(x) = \cos(x) \cdot x^2 + 2 \cdot \varepsilon, \quad x \in [0; 7; 0,01]$
9	$y(x) = \frac{1}{10+x^3} + 0,01 \cdot \varepsilon, \quad x \in [0; 20; 0,1]$
10	$y(x) = \frac{\sin(10x)}{x+0,1} + 0,5 \cdot \varepsilon, \quad x \in [0; 3; 0,01]$

### Содержание отчета

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Текст программы реализации метрического регрессионного метода на языке Python.
3. Результаты работы программы в виде графиков.
4. Выводы по полученным результатам.

## *Лабораторная работа №3*

### **Исследование работы алгоритма DBSCAN для задачи кластеризации данных**

**Цель работы:** реализовать и провести анализ работы алгоритма DBSCAN для решения задачи кластеризации заданной выборки.

### **Теоретический материал**

Теория для выполнения лабораторной работы доступна на следующих страницах сайта:

<https://proproprogs.ru/ml>

в разделах:

- Задачи кластеризации. Постановка задачи
- Алгоритм кластеризации DBSCAN

А также в соответствующих видеоматериалах, размещенных на странице сайта:

<http://tk.ulstu.ru/video.php?id=3>

### **Задания на лабораторную работу (по вариантам)**

1. Необходимо построить (реализовать на языке Python) алгоритм DBSCAN для кластеризации, следующих данных выборки (для своего варианта):

<http://tk.ulstu.ru/files/clusters.py>

2. Запустить алгоритм DBSCAN с разными значениями параметров  $\epsilon$  и  $m$ .

3. Отобразить результаты кластеризации в виде графика точек на плоскости (объекты разных классов должны быть иметь разные маркеры и цвет).

### **Содержание отчета**

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Текст программы алгоритма DBSCAN на языке Python.
3. Графики полученных результатов кластеризации.
4. Выводы по полученным результатам.



## *Лабораторная работа №4*

### **Реализация решающих деревьев для задач классификации**

**Цель работы:** научиться строить (обучать) решающие деревья применительно к задаче многоклассовой классификации.

#### **Теоретический материал**

Теория для выполнения лабораторной работы доступна на странице сайта:

<https://proproprogs.ru/ml>

в разделах:

- Логические методы классификации
- Критерии качества для построения решающих деревьев
- Построение решающих деревьев жадным алгоритмом ID3
- Усечение (pruning) дерева, обработка пропусков и категориальных признаков
- Решающие деревья в задачах регрессии. Алгоритм CART

а также в соответствующих видеоматериалах, размещенных на странице сайта:

<http://tk.ulstu.ru/video.php?id=3>

#### **Задания на лабораторную работу (по вариантам)**

1. Необходимо построить (реализовать на языке Python с применением библиотеки Scikit-Learn) решающее дерево для задачи классификации, следующих данных обучающей выборки (для своего варианта):

[http://tk.ulstu.ru/files/tree\\_data.py](http://tk.ulstu.ru/files/tree_data.py)

2. Решающие деревья строятся с использованием критерия Джини.

3. Исследуйте несколько вариантов решающих деревьев с разной максимальной глубиной (`max_depth`) и разным минимальным числом вершин в листах (`min_samples_leaf`).

4. Отобразите в виде графика на плоскости результаты деления двумерного признакового пространства полученными решающими деревьями.

#### **Содержание отчета**

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Программа, реализующая построение (обучение) решающего дерева по заданной обучающей выборке.
3. Графики и результаты деления признакового пространства решающими деревьями.
4. Выводы по полученным результатам.

## *Лабораторная работа №5*

### **Применение случайного леса (random forest) в задачах регрессии**

**Цель работы:** реализовать алгоритм random forest для аппроксимации функцией данных обучающей выборки.

#### **Теоретический материал**

Теория для выполнения лабораторной работы доступна на странице сайта:

<https://proproprogs.ru/ml>

в разделах:

- Логические методы классификации
- Критерии качества для построения решающих деревьев
- Построение решающих деревьев жадным алгоритмом ID3
- Усечение (pruning) дерева, обработка пропусков и категориальных признаков
- Решающие деревья в задачах регрессии. Алгоритм CART
- Случайные деревья и случайный лес. Бутстрэп и бэггинг

а также в соответствующих видеоматериалах, размещенных на странице сайта:

<http://tk.ulstu.ru/video.php?id=3>

#### **Задания на лабораторную работу (по вариантам)**

1. Необходимо реализовать на языке Python (с применением библиотеки Scikit-Learn) аппроксимацию заданной функцией случайным лесом (RandomForestRegressor).
2. Провести анализ восстановления функциональной зависимости при разных значениях параметров:
  - `n_estimators` – число используемых деревьев;
  - `max_depth` – максимальная глубина дерева.
3. Отобразить в виде графиков полученные результаты восстановления функции случайным лесом. (На каждом графике – координатных осях – должен быть исходный график и его результат восстановления алгоритмом random forest).

Вариант	Функция
1	$y(x) = \frac{\cos(20x)}{x+0,1} + 0,5 \cdot \varepsilon, \quad x \in [0; 4; 0,01]$
2	$y(x) = \sin(5x) \cdot \cos(x) + 0,1 \cdot \varepsilon, \quad x \in [0; 3; 0,01]$
3	$y(x) = \ln(x) \cdot e^{-x} + 0,01 \cdot \varepsilon, \quad x \in [1; 10; 0,01]$
4	$y(x) = \sin(x) \cdot x^2 + \varepsilon, \quad x \in [0; 3; 0,01]$
5	$y(x) = \sin(5x) \cdot e^{-x} + 0,1 \cdot \varepsilon, \quad x \in [0; 10; 0,01]$
6	$y(x) = \frac{5}{10+x^2} + 0,05 \cdot \varepsilon, \quad x \in [0; 20; 0,1]$
7	$y(x) = \cos(5x) \cdot e^x + 5 \cdot \varepsilon, \quad x \in [0; 4; 0,01]$
8	$y(x) = \cos(x) \cdot x^2 + 2 \cdot \varepsilon, \quad x \in [0; 7; 0,01]$
9	$y(x) = \frac{1}{10+x^3} + 0,01 \cdot \varepsilon, \quad x \in [0; 20; 0,1]$
10	$y(x) = \frac{\sin(10x)}{x+0,1} + 0,5 \cdot \varepsilon, \quad x \in [0; 3; 0,01]$

### Содержание отчета

1. Титульный лист с названием лабораторной работы, номером своего варианта, фамилией студента и группы.
2. Программа, реализующая алгоритм random forest.
3. Графики и результаты работы программы.
4. Выводы по полученным результатам.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Николенко С., Кадури́н А., Архангельская Е. Глубокое обучение. — СПб.: Питер, 2018. — 480 с.
2. Рашид, Тарик. Создаем нейронную сеть.: Пер. с англ. — СПб.: ООО «Альфа-книга», 2017. — 272 с.: ил.
3. Хайкин, Саймон. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. — М.: Издательский дом «Вильямс», 2006. — 1104 с.: ил.
4. Васильев К.К., Оптимальная обработка сигналов в дискретном времени: Учебн. пособие. — М.: Радиотехника, 2016. — 288 с.: ил.
5. Christopher M. Bishop, Neural Networks for Pattern Recognition. — Clarendon Press Oxford, 1995 - 498 с.
6. Harrison Kinsley, Neural Networks from Scratch in Python - 666 с.