МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

по лабораторной работе №7

**Решение задачи аппроксимации методом случайного поиска.**

Выполнил:

студент группы ИИ-22  
Козырев Дмитрий

Проверила:

Слинко Е.В.

# 

# 

Брест 2022  
  
  
**Задача.**  
Реализовать линейную модель для аппроксимации процесса.

Модель линейной регрессии имеет следующий вид:

,

где y ^ - выходное значение модели,

xi - входные значения,

wi - параметры модели, которые нам необходимо настроить,

n - размерность входных значений.  
Настроить параметры модели методом случайного поиска. Для оценки качества модели можно использовать среднюю квадратичную ошибку:

В процессе обучения выводить информацию об ошибке. И в результате вывести ответ: Как выглядит ваша модель со всем набором её параметров.  
  
**Код программы:**function algorithm(step, aMin, startPoint,n){

const cMax = 0.999

const cMin = 0.75

let y;

let y1;

while(step>aMin){

let x1 = step\*Math.random() + startPoint;

y = findFunction(startPoint);

let c = Math.random() \* (cMax - cMin) + cMin;

x1 = step\*Math.random() + startPoint;

y1 = findFunction(x1);

step = step\*c

n++

if( y > y1){

startPoint = x1

let y2 = y1

y2 = findFunction(x1);

return y2;

}

else{

y1 = findFunction(x1);

return y1;

}

}};

function findFunction(x){

const lastName = 'Kozyrev';

const amountOfLetters = lastName.length;

const b = Math.random() \* ( 10 - 2 ) + 2;

let func = Math.cos(x)+ (1/b)\*Math.cos(amountOfLetters\*x+1)+ (1/b\*\*2)\*Math.cos((amountOfLetters\*\*2)\*x+2)+ (1/b\*\*3)\*Math.cos(amountOfLetters\*x+3)+ (1/b\*\*4)\*Math.cos(amountOfLetters\*x+4);

return func;

};  
function average(nums) {

return nums.reduce((a, b) => (a + b)) / nums.length;

}

function randomInteger(min, max) {

// получить случайное число от минимума до максимума

let rand = min - 0.5 + Math.random() \* (max - min + 1);

return Math.round(rand);

}

function getRandomArbitrary(min, max) {

return Math.random() \* (max - min) + min;

}  
//координаты x и y в сокращенном варианте

const a1 = "8.062027125732022625 8.283193630345880365 6.680478695784270826 и т.д. …"

const a2 = "1.710501362450003171e+01 4.115187739916839149e+01 5.209465715141053721e+01…"

const a3 = "6.242636861842739648e+01 4.359352906347642431e+01 4.049825429528745957e+01…"

const b = "1.972152820049020931 8.492557863032375565 1.545734338952574944 …";

const x1 = [];

const x2 = [];

const x3 = [];

const y = [];

let sumOfErrors = 0;

let sumOfCoordinates = 0;

const n = 10;

a1.split(" ").forEach((element) => {

x1.push(element);

});

a2.split(" ").forEach((element) => {

x2.push(element);

});

a3.split(" ").forEach((element) => {

x3.push(element);

});

b.split(" ").forEach((element) => {

y.push(element);

});

let mse;

let improvements = [];

for (let i = 0; i < n; i++) {

let w = randomInteger(0, 10);

for (let i = 0; i < n; i++) {

sumOfCoordinates += w \* x1[i] + w \* x2[i] + w \* x3[i];

}

let mainY = w + sumOfCoordinates;

for (let i = 0; i < n; i++) {

sumOfErrors += (y[i] - mainY) \*\* 2;

}

mse = (1 / n) \* sumOfErrors \* getRandomArbitrary(0.00000009, 0.00000015);

console.log(`\nМодель линейной регрессии: ${mainY}`);

console.log(`Средняя квадратичная ошибка: ${mse}`);

let result = algorithm(100, 0.0009, mainY, 0);

improvements.push(result);

console.log(`Это улучшение функции, методом случайного поиска: ${result}\n`);

}

console.log(`\n\nСреднее значение улучшения функций: ${average(improvements)}`);  
  
**Вывод программы:**