Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

**Отчёт по лабораторной работе №1**

# Метод парзеновского окна с фиксированным размером окна

Выполнилa:

студент гр. ИП-813

Бурдуковский И.А.

Проверилa:

Морозова К.И.

Новосибирск 2021

Оглавление

[Задание 3](#_Toc52194686)

[Результаты 4](#_Toc52194687)

[Код программы 5](#_Toc52194688)

# Задание

Входные данные:

К заданию на лабораторную работу прилагаются файлы, в которых представлены наборы данных из объектов. Каждый объект описывается двумя признаками () и соответствующим ему классом ().

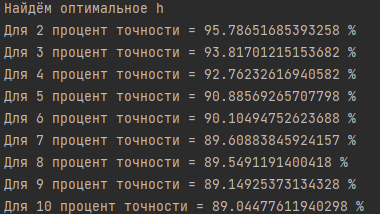
Суть лабораторной работы заключается в написании классификатора на основе метода k ближайших соседей. Данные из файла необходимо разбить на две выборки, обучающую и тестовую, согласно общепринятым правилам разбиения. На основе этих данных необходимо обучить разработанный классификатор и протестировать его на обеих выборках. В качестве отчёта требуется представить работающую программу и таблицу с результатами тестирования для каждого из 10 разбиений. Разбиение выборки необходимо выполнять программно, случайным образом, при этом, не нарушая информативности обучающей выборки. Разбивать рекомендуется по следующему правилу: делим выборку на 3 равных части, 2 части используем в качестве обучающей, одну в качестве тестовой. Кроме того, обучающая выборка должна быть сгенерирована таким образом, чтобы минимизировать разницу между количеством представленных в ней объектов разных классов, т.е. .

Функция ядра:

Файл с данными - 2

# Результаты

Ход вычисления H:



Самый оптимальный результат получается при ширине окна равном двум. При таком размере окна на тестовой выборке вероятность правильной классификации равна 95,8%.

# Код программы

import csv  
from math import sqrt  
from sklearn.model\_selection import train\_test\_split  
  
  
def loo(X\_train, h\_max: int = 10):  
 h\_calc = 0  
 h\_accuracy = 0  
  
 for h in range(2, h\_max + 1):  
 correct = 0  
 incorrect = 0  
  
 for dot in X\_train:  
 class\_0 = 0  
 class\_1 = 0  
  
 for influence\_dot in X\_train:  
 if not dot == influence\_dot:  
 if core(distance(dot, influence\_dot) / h) <= 1:  
 if influence\_dot[2] == 0:  
 class\_0 += 1  
 else:  
 class\_1 += 1  
  
 if class\_0 + class\_1 == 0:  
 continue  
  
 if class\_0 >= class\_1:  
 calc = 0  
 else:  
 calc = 1  
  
 # print(calc)  
  
 if dot[2] == calc:  
 correct += 1  
 else:  
 incorrect += 1  
  
 accuracy = correct / (correct + incorrect)  
 print("Для {0} процент точности = {1}".format(h, accuracy \* 100), "%")  
 if h\_accuracy < accuracy:  
 h\_accuracy = accuracy  
 h\_calc = h  
 return h\_calc  
  
  
def parzen(X\_test, h):  
 correct = 0  
  
 for dot in X\_test:  
 class\_0 = 0  
 class\_1 = 0  
 for influence\_dot in X\_test:  
 if not (dot[0] == influence\_dot[0] and dot[1] == influence\_dot[1]):  
 if core(distance(dot, influence\_dot) / h) <= 1:  
 if influence\_dot[2] == 0:  
 class\_0 += 1  
 else:  
 class\_1 += 1  
 if class\_0 > class\_1:  
 calc = 0  
 else:  
 calc = 1  
  
 if dot[2] == calc:  
 correct += 1  
  
 return correct  
  
  
def distance(central\_dot, influence\_dot):  
 return sqrt((central\_dot[0] - influence\_dot[0]) \*\* 2 + (central\_dot[1] - influence\_dot[1]) \*\* 2)  
  
  
def core(r):  
 return abs(1 - r)  
  
  
X\_data = []  
Y\_data = []  
with open("data2.csv") as f:  
 csv\_iter = csv.reader(f, delimiter=',')  
 next(csv\_iter)  
 for row in csv\_iter:  
 X\_data.append(row)  
  
for row in X\_data:  
 row[0] = int(row[0])  
 row[1] = int(row[1])  
 row[2] = int(row[2])  
  
X\_train, X\_test = train\_test\_split(X\_data, test\_size=0.33)  
  
print("Найдём оптимальное h")  
h = loo(X\_train, 10)  
print(h)  
acc = parzen(X\_test, h)  
print("Точность на тестовой выборке {0}".format(acc / len(X\_test) \* 100), "%")