

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ЛАБОРАТОРНАЯ РАБОТА № 1  
Метод k ближайших соседей.

Выполнил:  
Студент группы: ИП-715  
Винтер Антон

Проверил: ассистент кафедры ПМиК  
Морозова К.И.

## **Оглавление**

1. Текст задания
2. Результат работы программы
3. Код программы

### Текст задания.

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.  $abs(|\{(x_i, y_i) \in X^l | y_i = -1\}| - |\{(x_i, y_i) \in X^l | y_i = 1\}|) \rightarrow \min$ .

- В нашем варианте используем классификатор: “Метод парзеновского окна с фиксированным  $h$ ”
- Так же использовали функцию ядра  $K(z)$ :

“Q-квадратическое  $K(x) = (1 - r^2)^2 [r \leq 1]$ ”

- Кроме того использовали файл с данными для классификации номер 4 (data4.csv)

## Результат работы программы

---

H =	1	98.6159169550173 %
H =	2	95.14360313315927 %
H =	3	92.55517149694231 %
H =	4	90.84658358401376 %
H =	5	89.84609609609609 %
H =	6	89.37138854841534 %
H =	7	88.38919288645691 %
H =	8	87.8680800942285 %
H =	9	87.45609633718013 %
H =	10	87.57307499582429 %
H =	11	88.18818818818819 %
H =	12	88.34028356964137 %
H =	13	88.59619873291096 %
H =	14	88.71478579763293 %
H =	15	88.84814135689282 %
H =	16	89.16486081013502 %
H =	17	89.28154692448742 %
H =	18	89.29821636939491 %
H =	19	89.29821636939491 %
H =	20	89.36489414902485 %
h_opt =	1	
Result =		79.8599533177726 %

---

## Код программы

`#@title Текст заголовка по умолчанию`

```
import csv
import random
import math

h = 0.05
test_perc = 0.3

class_0 = []
class_1 = []

test = []
study = []

def LOO(st,tst):
    loo_list = st[:3000]
    loo_list.extend(tst)

    h_opt = 0
    perc_opt = 0.0

    k = 1
    while k<=20:
        good_answ = 0
        bad_answ = 0

        for i in range(0,len(loo_list)):
            vote1 = 0
            cnt = 0
            vote0 = 0
            for j in range(0,len(loo_list)):
                if (i!=j):
                    if (core(dist(loo_list[i],loo_list[j])/k)<=1):
                        cnt+=1
                        if (loo_list[j][2]==0):
                            vote0+=1
                        else:
                            vote1+=1

            if (vote1+vote0==0):
                continue
            if (vote1>=vote0):
                answ_class = 1
            else:
                answ_class = 0

            if (loo_list[i][2]==answ_class):
                good_answ+=1
            else:
```

```
bad_answ+=1
```

```
print("H = ",k," ",(good_answ/(good_answ+bad_answ))*100,"%")
```

```
if perc_opt<((good_answ/(good_answ+bad_answ))*100):  
    perc_opt = (good_answ/(good_answ+bad_answ))*100  
    h_opt = k
```

```
k+=1
```

```
print("h_opt = ", h_opt)
```

```
return h_opt
```

```
def core(r):
```

```
    return math.pow(1 - math.pow(r, 2), 2)
```

```
def dist(a,b):
```

```
    return math.sqrt(math.pow(a[0]-b[0],2)+math.pow(a[1]-b[1],2))/1.0
```

```
def main():
```

```
    csv_path = "data4.csv"
```

```
    with open(csv_path, "r") as f_obj:
```

```
        csv_reader(f_obj)
```

```
    cnt = 0
```

```
class_1.pop(0)
```

```
random.shuffle(class_0)
```

```
random.shuffle(class_1)
```

```
for i in range(0,len(class_0)):
```

```
    class_0[i] = [int(class_0[i][0]),int(class_0[i][1]),int(class_0[i][2])]
```

```
for i in range(0,len(class_1)):
```

```
    class_1[i] = [int(class_1[i][0]),int(class_1[i][1]),int(class_1[i][2])]
```

```
c = int(len(class_0)*test_perc)
```

```
test = class_0[:c]
```

```
study = class_0[c:]
```

```
c = int(len(class_1)*test_perc)
```

```
test.extend(class_1[:c])
```

```
study.extend(class_1[c:])
```

```
h = LOO(study,test)
```

```
class_0_votes = 0
```

```
class_1_votes = 0
```

```
correct = 0
```

```

for i in test:
    for j in study:
        if core(dist(i,j)/h) <= 1:
            if j[2] == 0:
                class_0_votes = class_0_votes + 1
            else:
                class_1_votes = class_1_votes + 1

    if (class_0_votes >= class_1_votes) and i[2]==0:
        correct = correct + 1
    elif (class_0_votes < class_1_votes) and i[2]==1:
        correct = correct + 1

```

```

print("Result = ",(correct/len(test))*100,"%")

```

```

def csv_reader(file_obj):
    reader = csv.reader(file_obj)

    for row in reader:
        if row[2] == "0":
            class_0.append(row)
        else:
            class_1.append(row)

if __name__ == "__main__":
    main()

```