**Отчет по дисциплине:**  Теория машинного обучения

**Язык программирования:** Python

**Тема №4:** Метод Lowess для непараметрической регрессии

**Ход работы программы:**

Алгоритм Lowess является улучшением формулы Надарая-Ватсона. Для каждого объекта выборки вводится вес , обозначающий значение этого объекта для алгоритма.

Выпишем формулу LOWESS, где - ширина окна:

def lowess(x,X,Y,weights,kernel,h):

if h == None:

h = 1

upper\_sum = 0

down\_sum = 0

for i in range(len(X)):

upper\_sum = upper\_sum + Y[i]\*weights[i]\*kernel(metrics.pairwise.euclidean\_distances([x],[X[i]])[0][0]/h)

down\_sum = down\_sum + weights[i]\*kernel(metrics.pairwise.euclidean\_distances([x],[X[i]])[0][0]/h)

if(upper\_sum==0):

return 0

return upper\_sum/down\_sum

Для того, чтобы найти оптимальные значения весов применяется **алгоритм cross validation:**

def cross\_validation(X,Y,kernel,steps,h):

weights = [1 for x in X]

errors = [1 for x in X]

for i in range(steps):

for k in range(len(X)):

item = X[k]

val = Y[k]

X\_t = np.delete(X,k,0)

Y\_t = np.delete(Y,k)

weights\_t = np.delete(weights,k)

val\_t = lowess(item,X\_t,Y\_t,weights\_t,kernel,h)

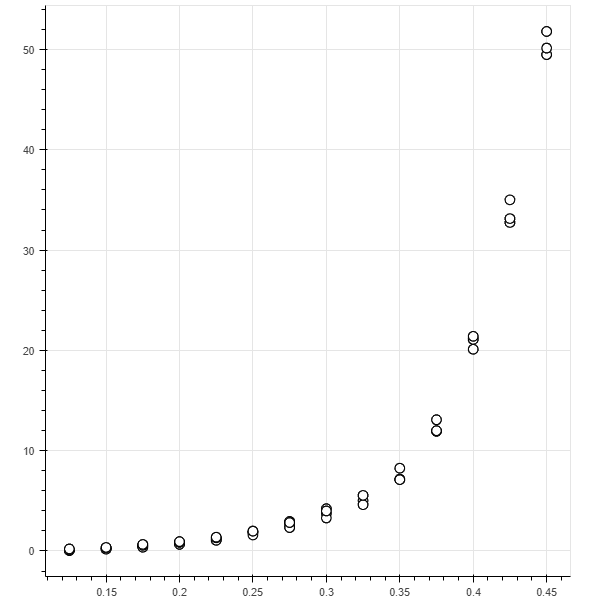
errors[k] = abs(val\_t-val)

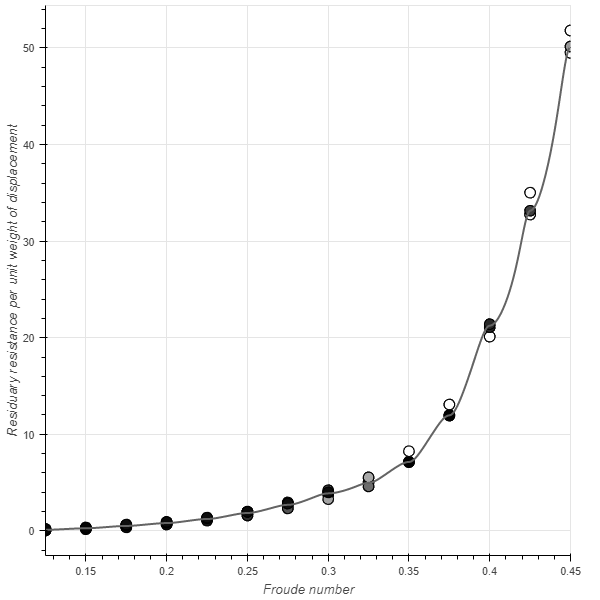
s = np.median(errors)

weights = [(1-abs(e/(6\*s))\*\*2)\*\*2 if abs((e/(6\*s)))<=1 else 0 for e in errors]

return weights

Для данного алгоритма выбрана выборка, где в качестве параметров различные свойства кораблей. В качестве множества **Y** выберем значение водоизмещения корабля.

Считать будем относительно параметра «Froude number».

Результат при использовании квадратического ядра: 

Результат при использовании Гауссова ядра:

