

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

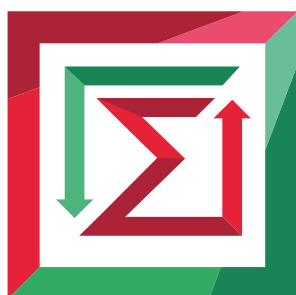


**НГТУ  
НЭТИ**

Кафедра теоретической и прикладной информатики

Лабораторная работа №3  
по дисциплине «Основы теории машинного обучения»

**Построение регрессионных моделей по методу LS SVM**



ФАКУЛЬТЕТ: ПМИ

ГРУППА: ПМИМ-01

СТУДЕНТЫ: Ершов П.К.  
Малышкина Е.Д.  
Слободчикова А.Э.

ВАРИАНТ: 4

ПРЕПОДАВАТЕЛЬ: Попов А.А.

Новосибирск

2021

## 1. Цель.

Получить практические навыки по построению регрессионных зависимостей с использованием метода опорных векторов с квадратичной функцией потер.

## 2. Содержание работы.

1. Ознакомиться с теоретическими основами метода опорных векторов в модификации, связанной с использованием квадратичной функции потерь (LS SVM).
2. Провести моделирование экспериментальных данных в соответствии с вариантом задания.
3. По полученным экспериментальным данным построить регрессионную модель по методу LS SVM.
4. Оформить отчет, включающий в себя постановку задачи, полученные результаты и выводы.
5. Защитить лабораторную работу.

## 3. Исходные данные.

Вариант задания:

Моделируемая зависимость одномерная:  $u(x) = x - 3e^{-\frac{(x)^2}{0,15}} + 3e^{-\frac{(x-1)^2}{0,1}}$

Интервал варьирования фактора  $x \in [-1; 2]$ .

Применяется гауссово ядро вида:  $K(x, z) = e^{-\frac{\|x-z\|^2}{2\sigma^2}}$ .

#### 4. Исследование:

По теореме Мерсера, результирующая LS-SVM модель для оценки функции имеет следующий:

$$\hat{m}_n(x) = \sum_{k=1}^n \hat{a}_k K(x, x_k) + \hat{b}, \text{ где}$$

$$\hat{a} = \left( \Omega + \frac{1}{\gamma} I_n \right)^{-1} (y - i_n \hat{b}),$$

$$\hat{b} = \frac{i_n^T \left( \Omega + \frac{1}{\gamma} I_n \right)^{-1} y}{i_n^T \left( \Omega + \frac{1}{\gamma} I_n \right)^{-1} i_n}$$

$K(x, x_k)$  – функция, удовлетворяющая условию Мерсера, называемая ядром.

Сгенерируем 100 наблюдений с помощью равномерного распределения с добавлением шума Гаусса.

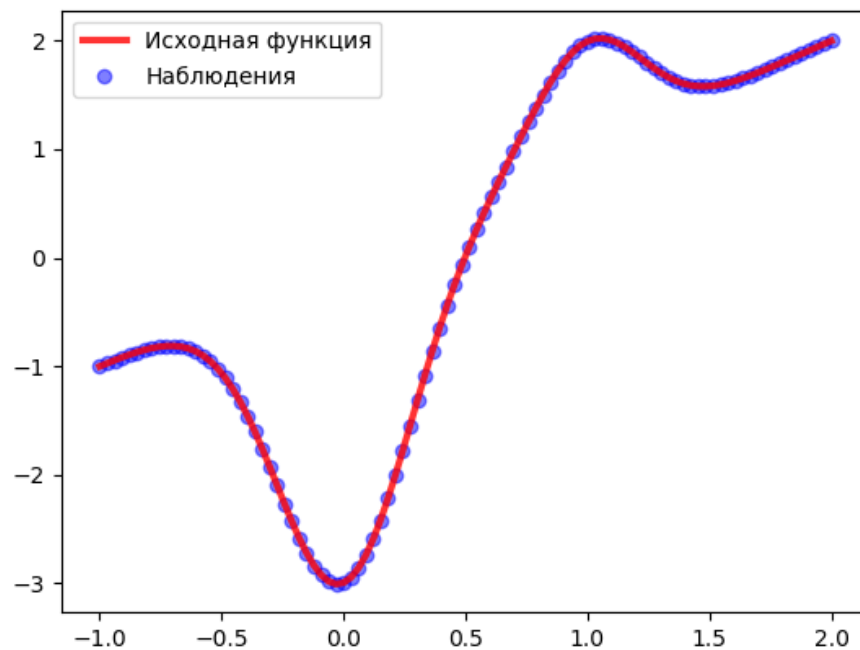


Рисунок 1. График функции и наблюдений без добавления шума

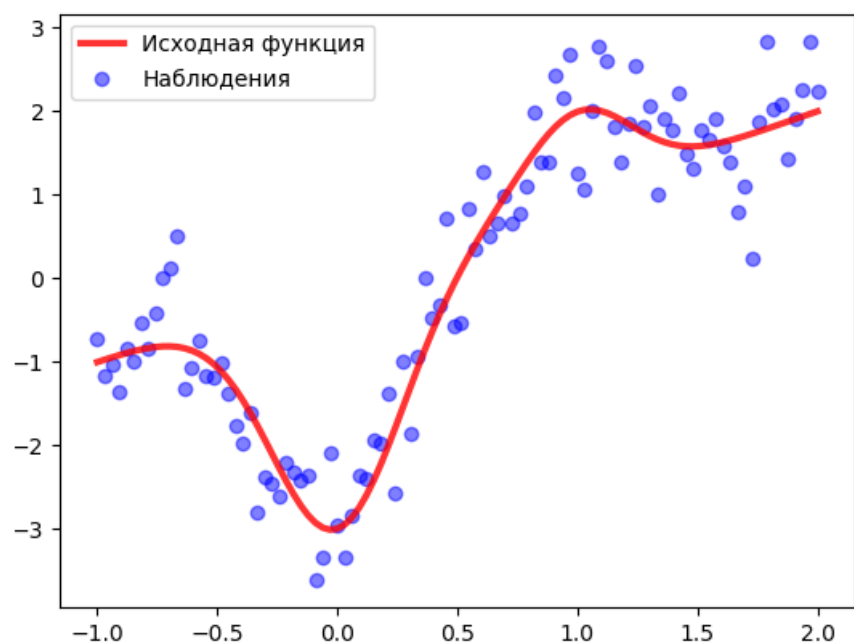


Рисунок 2. График функции и наблюдений с добавлением шума

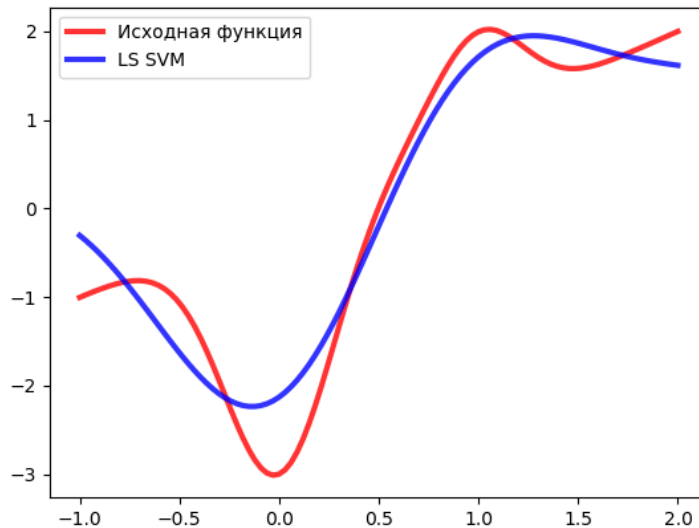
Таблица1. Значения параметра, истинных наблюдений и наблюдений с шумом.

$x$	$y_{\text{истинное}}$	$y_{\text{шум}}$
-1,0000	-1,0038	-1,2402
-0,9697	-0,9754	-0,2463
-0,9394	-0,9478	-1,2094
-0,9091	-0,9212	-0,2248
-0,8788	-0,8962	-1,4126
-0,8485	-0,8732	-0,4926
-0,8182	-0,8528	-1,6816
-0,7879	-0,8357	-1,2954
-0,7576	-0,8230	-1,5857
-0,7273	-0,8155	-1,6009
-0,6970	-0,8146	-1,1656
-0,6667	-0,8217	-1,0381
-0,6364	-0,8380	-1,2571
-0,6061	-0,8653	-0,3429
-0,5758	-0,9049	-1,1373
-0,5455	-0,9582	-0,8222
-0,5152	-1,0266	-1,1359
-0,4848	-1,1107	-1,5040
-0,4545	-1,2112	-0,4081
-0,4242	-1,3279	-1,4662
-0,3939	-1,4601	-2,2526
-0,3636	-1,6061	-1,6765
-0,3333	-1,7636	-1,6532
-0,3030	-1,9295	-2,1039
-0,2727	-2,0999	-2,3562
-0,2424	-2,2699	-2,4645
-0,2121	-2,4346	-2,5360
-0,1818	-2,5884	-2,6203
-0,1515	-2,7258	-3,5878
-0,1212	-2,8413	-2,7629
-0,0909	-2,9301	-3,6993
-0,0606	-2,9880	-2,1533
-0,0303	-3,0120	-3,4918
0,0000	-3,0000	-3,2589
0,0303	-2,9513	-3,2584

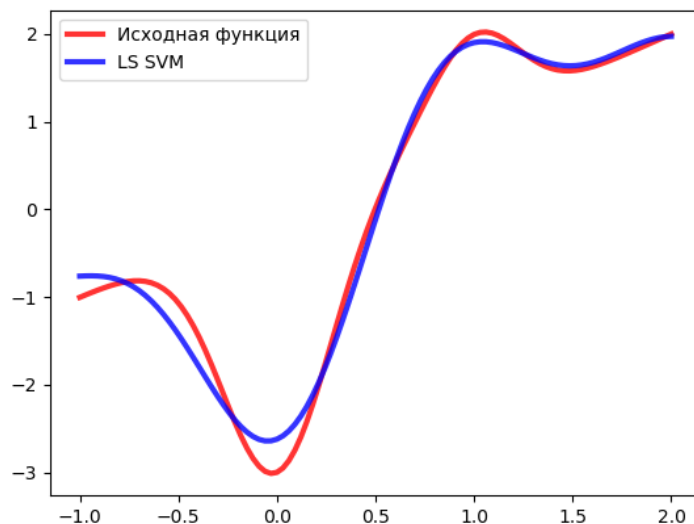
0,0606	-2,8667	-2,3240
0,0909	-2,7480	-3,1406
0,1212	-2,5984	-2,5731
0,1515	-2,4220	-2,0864
0,1818	-2,2236	-2,5200
0,2121	-2,0084	-1,6720
0,2424	-1,7819	-0,8707
0,2727	-1,5494	-2,4619
0,3030	-1,3157	-1,4130
0,3333	-1,0852	-0,9248
0,3636	-0,8614	-0,4431
0,3939	-0,6468	-1,0056
0,4242	-0,4431	-0,9428
0,4545	-0,2511	-0,6296
0,4848	-0,0707	-0,5217
0,5152	0,0990	0,1953
0,5455	0,2594	0,8400
0,5758	0,4120	-0,6705
0,6061	0,5587	0,2855
0,6364	0,7012	0,7785
0,6667	0,8409	1,4380
0,6970	0,9785	1,2396
0,7273	1,1143	0,3208
0,7576	1,2478	0,8823
0,7879	1,3777	1,5978
0,8182	1,5021	1,7648
0,8485	1,6187	1,8177
0,8788	1,7247	1,0951
0,9091	1,8176	1,7517
0,9394	1,8950	1,6703
0,9697	1,9549	2,2910
1,0000	1,9962	2,2707
1,0303	2,0186	2,6436
1,0606	2,0229	2,1924
1,0909	2,0105	2,9741
1,1212	1,9839	2,2810
1,1515	1,9460	0,7564
1,1818	1,9001	1,4685
1,2121	1,8496	2,6268
1,2424	1,7979	1,4287
1,2727	1,7480	1,5633
1,3030	1,7022	1,0932
1,3333	1,6625	2,0547
1,3636	1,6301	2,1666
1,3939	1,6058	1,8414
1,4242	1,5896	1,8860
1,4545	1,5812	3,2579
1,4848	1,5801	1,1458
1,5152	1,5855	0,7429
1,5455	1,5965	2,1875
1,5758	1,6121	1,3969
1,6061	1,6315	1,3676
1,6364	1,6538	1,9408
1,6667	1,6784	2,1218
1,6970	1,7047	1,6227
1,7273	1,7323	1,7094
1,7576	1,7608	1,4541
1,7879	1,7899	0,3509
1,8182	1,8194	1,4580
1,8485	1,8492	2,9106
1,8788	1,8792	2,0448
1,9091	1,9093	2,1705
1,9394	1,9395	2,5024
1,9697	1,9698	2,7153
2,0000	2,0000	1,7266

Оптимальные параметры  $\sigma$  и  $\gamma$  будем находить перебором. В качестве оценки модели будем использовать среднеквадратическую ошибку (далее MSE) и коэффициент детерминации (далее  $R^2$ ). Первый параметр показывает величину ошибки, а второй достоверность прогнозов, где 1 абсолютно достоверный прогноз, а 0 абсолютно недостоверный.

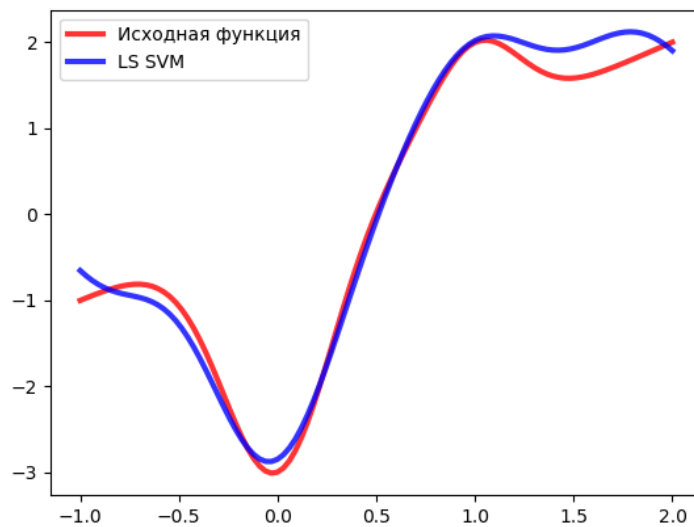
$\sigma$	$\gamma$	MSE	$R^2$
1	1	0.14862922	0.94947331



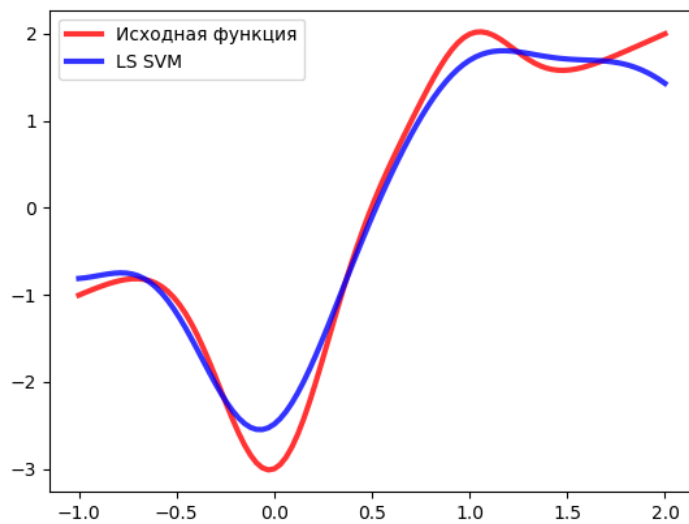
$\sigma$	$\gamma$	MSE	$R^2$
2	2	0.02895229	0.99015763



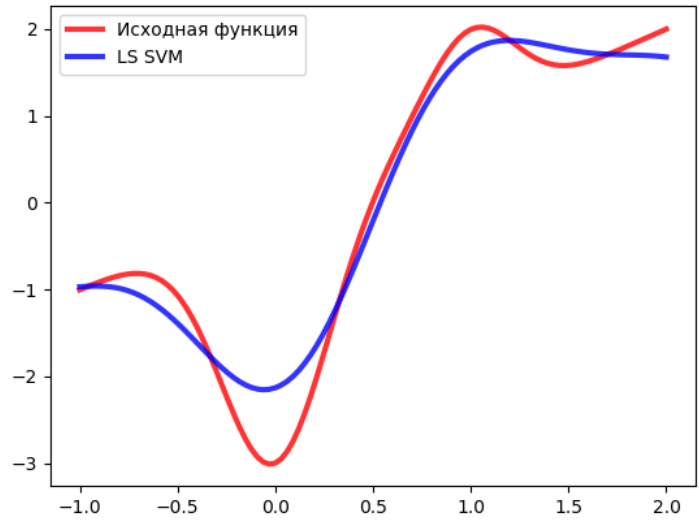
$\sigma$	$\gamma$	MSE	$R^2$
4	4	0.03520103	0.98803337



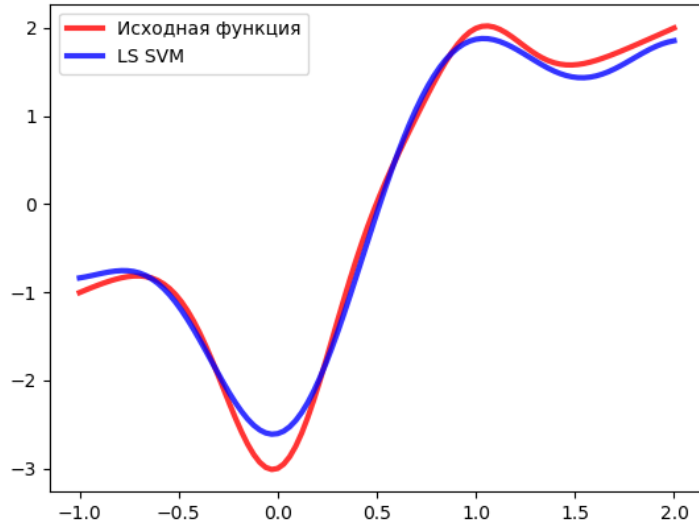
$\sigma$	$\gamma$	MSE	$R^2$
3	3	0.05265534	0.98209975



$\sigma$	$\gamma$	MSE	$R^2$
1,5	1,5	0.10311821	0.96494484

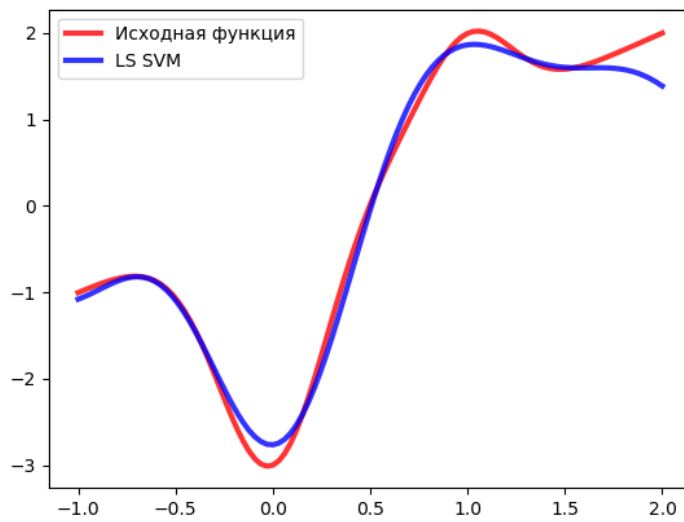


$\sigma$	$\gamma$	MSE	$R^2$
2,5	2,5	0.02418536	0.99177816

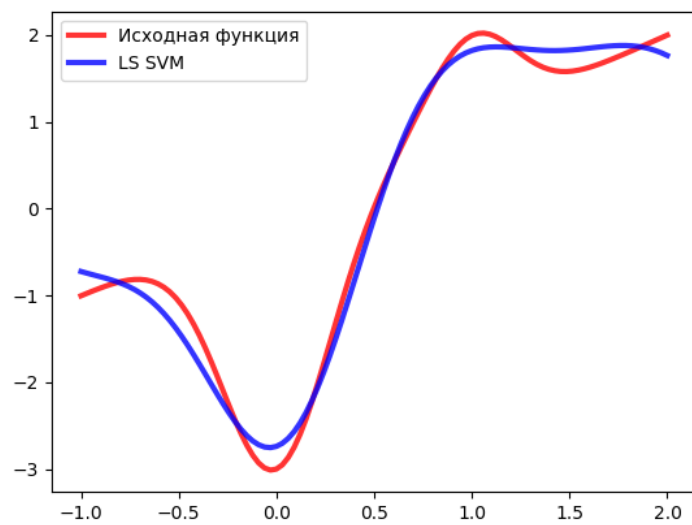




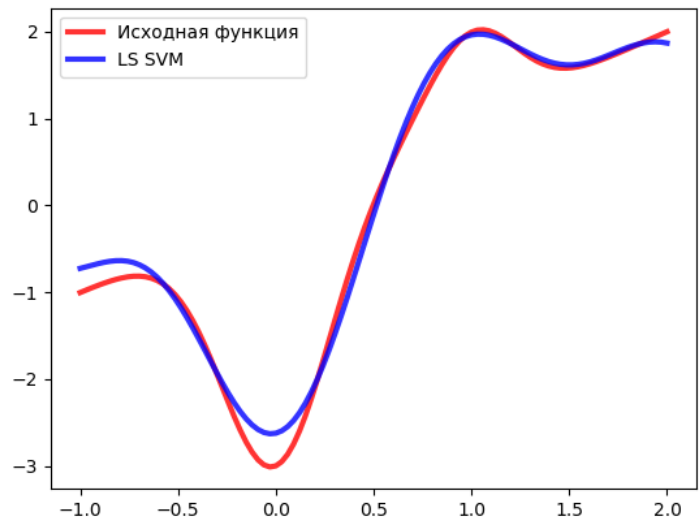
$\sigma$	$\gamma$	MSE	$R^2$
2,7	2,7	0.02657724	0.99096503



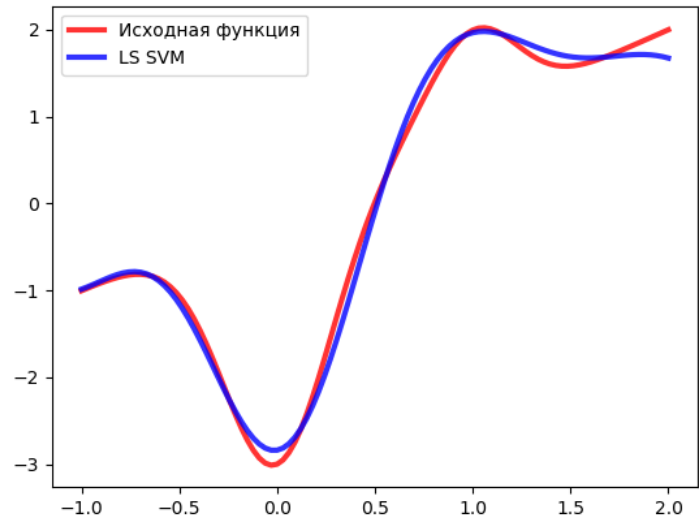
$\sigma$	$\gamma$	MSE	$R^2$
2,6	2,6	0.03230127	0.98901914



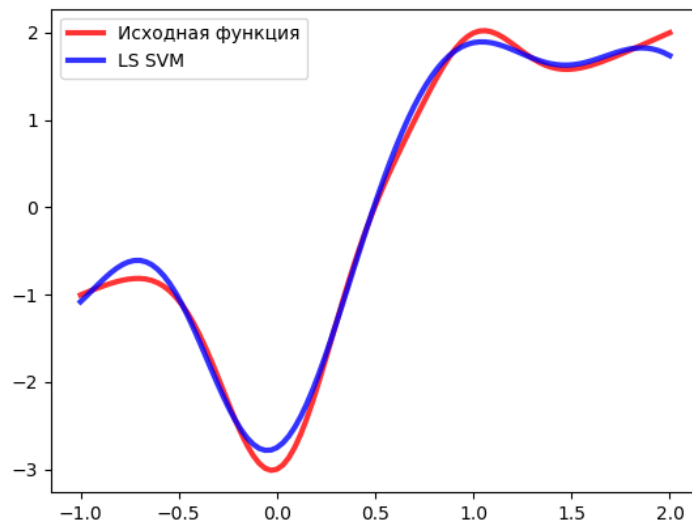
$\sigma$	$\gamma$	MSE	$R^2$
2,54	2,56	0.02265689	0.99229776



$\sigma$	$\gamma$	MSE	$R^2$
2,6	2,56	0.01552899	0.99472090



$\sigma$	$\gamma$	MSE	$R^2$
2,6	2,54	0.01337817	0.99545208



Таким образом, наиболее оптимальными параметрами являются  $\sigma = 2,6$  и  $\gamma = 2,54$ . При данных параметрах среднеквадратическая ошибка наименьшая, а коэффициент детерминации ближе всего к 1.

## 5. Текст программы:

```
import numpy as np
import pandas as pd
import math
from sklearn.svm import SVC
import matplotlib.pyplot as plt
from sklearn.metrics.pairwise import rbf_kernel
from sklearn.metrics import r2_score, mean_squared_error

# Число наблюдений
size = 100

sigma = 2.6
gamma = 2.54

# Моделируемая функция
def u(x):
    return x - 3 * np.exp(-1 * np.power(x, 2) / 0.15) + np.exp(-1 * np.power(x - 1, 2) / 0.1)

# Гауссово ядро
def RBF(x):
    K = 1.0 * rbf_kernel(x, gamma = sigma)
    return K

# Вычисление предсказания по модели
def predict(alfa, x, beta):
    return np.dot(alpha, RBF(x)) + beta

# Интервал фактора X
l_b = -1
h_b = 2

# Генерируем массив наблюдения
x = np.linspace(l_b, h_b, size).reshape(-1, 1)

# Вычисляем значения функции
y_list = u(x)

# Добавляем шумы
gaussian_noise = np.random.normal(0, 0.5, size).reshape(-1,1)
y = y_list + gaussian_noise

print('x=\n', x)
print('y_list=\n', y_list)
print('y=\n', y)

df = pd.DataFrame({'x': x.ravel(), 'y_list': y_list.ravel(), 'y': y.ravel()})
df.to_excel("D:/Магистратура/Попов/lab3/lab3.xlsx")

plt.scatter(x, y_list, c='b', alpha=0.5, label="Наблюдения")
plt.plot(x, y_list, 'r', lw=3, label='Исходная функция', alpha=0.8)
plt.legend()
plt.show()

plt.scatter(x, y, c='b', alpha=0.5, label="Наблюдения")
plt.plot(x, y_list, 'r', lw=3, label='Исходная функция', alpha=0.8)
plt.legend()
plt.show()

y = y[:, np.newaxis]
Ones = np.array([[1]] * len(y))
K = RBF(x)
A_cross = np.linalg.pinv(np.block([
    [0, Ones.T],
```

```

    [Ones, K + 1.0 / gamma * np.identity(len(y))]
    ]))

B = np.concatenate(([0], y), axis=None)

solution = np.dot(A_cross, B)

beta = solution[0]
alpha = solution[1:]
print('beta=\n', beta)
print('alpha=\n', alpha)
print('K=\n', K)

# Вычисление y по модели
y_predict = predict(alpha, x, beta)
print('y_predict=\n', y_predict)

plt.plot(x, y_ist, 'r', lw=3, label='Исходная функция', alpha=0.8)
plt.plot(x, y_predict, 'b', lw=3, label='LS SVM', alpha=0.8)
plt.legend()
plt.show()

mse = mean_squared_error(y_ist, y_predict)
r2 = r2_score(y_ist, y_predict)
print('MSE=\n', format(mse, '.8f'))
print('R2=\n', format(r2, '.8f'))

```