|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
|  | | |
| Лабораторная работа №3 | | |
| по дисциплине «Статистические методы анализа данных» | | |
| **ИНТЕРВАЛЬНОЕ ОЦЕНИВАНИЕ, ПРОВЕРКА ГИПОТЕЗ И ПРОГНОЗИРОВАНИЕ.** | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-12 |
| Бригада: | 10 |
| Студенты: | Швадченко Артём Cубботин Дмитрий |
|  |
| Преподаватель: | Попов Александр Александрович |
|  | | |
| Новосибирск | | |
|  | | |
| 2024 | | |

Цель:

Изменить модель регрессии, добавив в нее дополнительный регрессор, ранее не вошедший в состав модели, порождающей данные. Не генерируя новых данных, найти точечные оценки всех параметров расширенной модели. В дальнейшем при рассмотрении этой расширенной модели анализе должно быть показано, что параметр при дополнительном регрессоре незначим.

Построить доверительные интервалы для каждого параметра модели регрессии.

Проверить гипотезу о незначимости каждого параметра модели.

Проверить гипотезу о незначимости самой регрессии.

Рассчитать прогнозные значения для математического ожидания функции отклика x,=fT(x) для всего интервала действия одного из факторов, зафиксировав значения других факторов на границе или в центре области их определения.

По полученным в п. 5 прогнозным значениям построить графики прогнозных значений и доверительной полосы для математического ожидания функции отклика и для самого отклика.

Заново смоделировать исходные данные (см. лаб. работу № 1), увеличив мощность случайной помехи до 50…70 % от мощности полезного сигнала, и провести оценку параметров. Повторить пункты 3, 4 с новыми данными.

Ход работы:

1. Добавили в модель регрессор x1\*\*3 с коэффициентом:

Применив код для оценки параметров, написанный в ходе лабораторной работы №2, получили следующий вектор оценок параметров:

[ 1.07 -0.01 0.95 0.13 0.97 0.08]

Действительно, оценка коэффициента при дополнительном регрессоре оказалась незначительной, тогда как остальные оценки близки к полученным в лабораторной работе №2.

1. Построим доверительные интервалы для каждого параметра. Так как в выборке всего 25 значений, для этого будем применять t-распределение (распределение Стьюдента) из библиотеки scipy. Для расчёта интервалов использовали следующее неравенство:

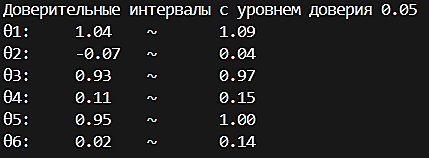
где ; квантиль распределения Стьюдента;

Для уровня доверия 90% получили следующие интервалы:

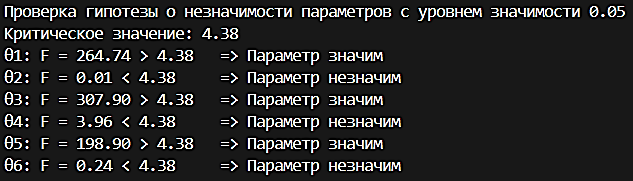


Из-за высокого уровня доверия интервалы оказались довольно большими, но все истинные значения параметров попадают в них.

Если же установить уровень доверия 5%, то доверительные интервалы становятся значительно меньше, но истинные значения для параметров , в них не попали.



1. Для каждого из параметров проверили гипотезу о незначимости:



Гипотеза о незначимости параметров подтвердилась только для тех параметров, которые изначально задавались как незначимые, а также для параметра при добавленном регрессоре, то есть проверка соответствует модели, по которой генерировалась выборка.

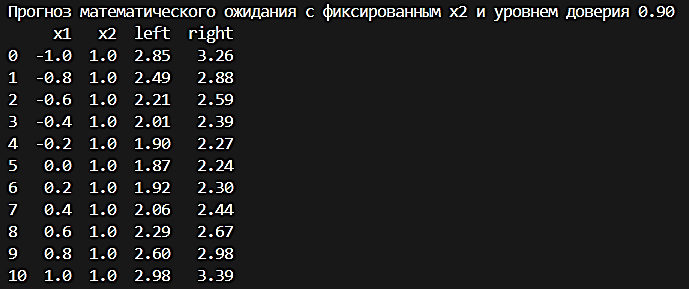
1. Проверили гипотезу о незначимости регрессии:

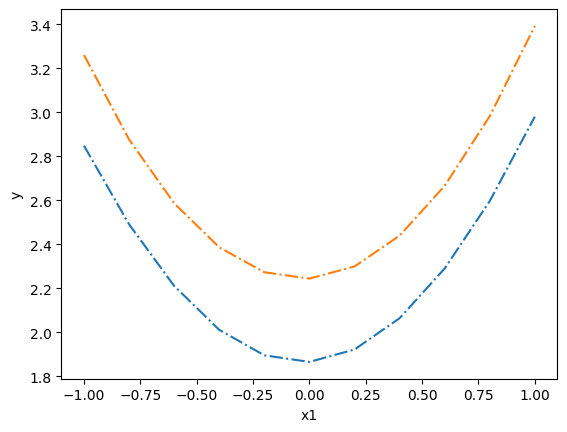
где и

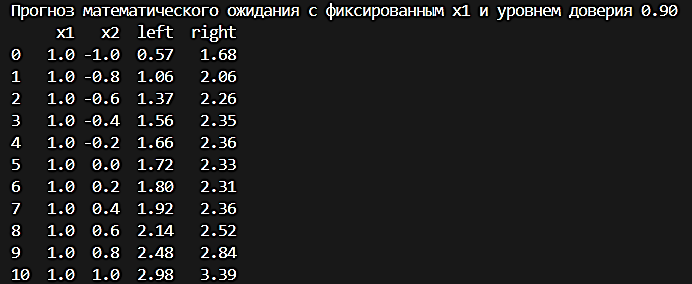


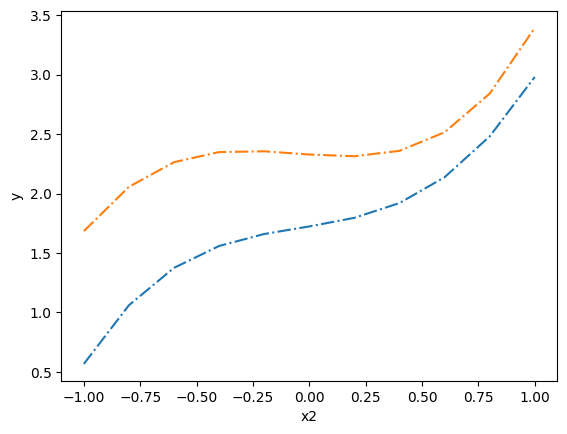
5-6. Для расчёта прогнозных значений математического ожидания отклика воспользуемся неравенством:

где ,

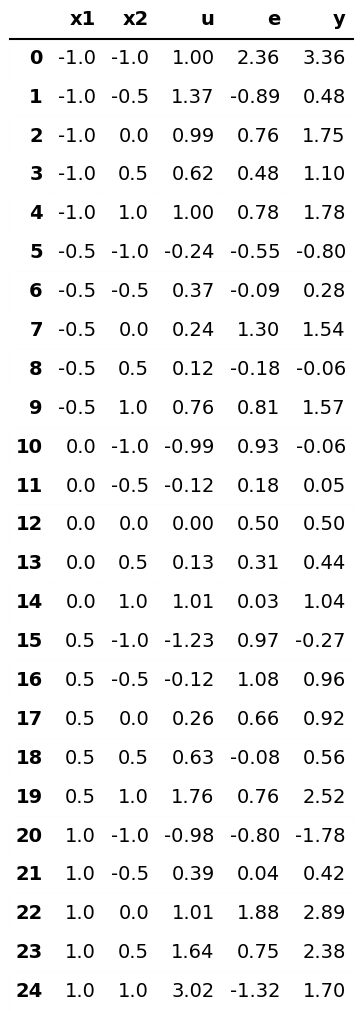


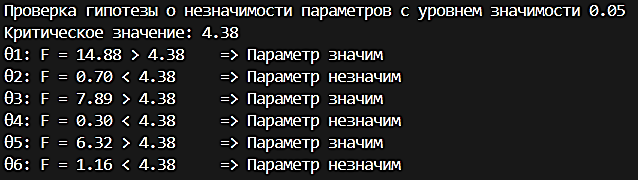






7. С помощью кода из лабораторной работы №1 сгенерировали новый набор данных, повысив долю дисперсии от мощности сигнала до 70%.





Cтатистики значимых параметров стали значительно меньше, т.е. гипотеза о их незначимости отвергается с меньшей уверенностью.

Проверили гипотезу о незначимсоти регрессии. Гипотеза всё ещё отвергается при заданном уровне значимости, но значение статистики стало на порядок меньше. То есть как и при проверке незначимости параметров, уверенность в результате уменьшилась.



**Код программы**

import numpy as np

import pandas as pd

import scipy.stats as sp

import matplotlib

import matplotlib.pyplot as plt

matplotlib.use('TkAgg')

significance = 0.05

alpha = 0.1

if \_\_name\_\_ == '\_\_main\_\_':

# Считывание данных

data = pd.read\_csv("c:/Users/artem/Documents/Учёба/7 сем/СМАд/data.csv")

# Формируем матрицу X по модели с дополнительным регрессором x1\*\*3.

X = np.array([[data.loc[i, "x1"] \*\* 2,

data.loc[i, "x1"],

data.loc[i, "x2"] \*\* 3,

data.loc[i, "x2"] \*\* 2,

data.loc[i, "x1"] \* data.loc[i, "x2"],

data.loc[i, "x1"] \*\* 3

]

for i in data.index])

# Производим оценку параметров

theta\_vector = np.linalg.inv(X.T @ X) @ X.T @ data["y"].to\_numpy()

print("Вектор точечных оценок параметров: ", theta\_vector.round(2))

data["y^"] = (X @ theta\_vector).tolist()

# Построение доверительных интервалов для каждого параметра

temp = np.linalg.inv(X.T @ X)

print("Доверительные интервалы с уровнем доверия {:.2f}".format(1-alpha))

for j in range(len(theta\_vector)):

theta\_dev = abs(np.sqrt(temp[j][j]) \* sp.t.ppf(alpha/2, len(X)-len(theta\_vector)))

print("θ{:d}:\t{:.2f}\t~\t{:.2f}".format(j+1, theta\_vector[j]-theta\_dev, theta\_vector[j]+theta\_dev))

# Производим оценку ошибок

error\_est = data["y"].to\_numpy() - X @ theta\_vector

data["y-y^"] = error\_est.tolist()

# Производим оценку дисперсии

dispersion\_est = (error\_est.T @ error\_est) / (len(X)-len(theta\_vector))

# Проверка о незначимости параметров

print("Проверка гипотезы о незначимости параметров с уровнем значимости {:.2f}".format(significance))

F\_critical = sp.f.isf(significance, 1, len(X) - len(theta\_vector))

print("Критическое значение: {:.2f}".format(F\_critical))

for j in range(len(theta\_vector)):

F = (theta\_vector[j]\*\*2/(dispersion\_est \* temp[j][j]))

if F < F\_critical:

print("θ{:d}: F = {:.2f} < {:.2f} \t=> Параметр незначим".format(j + 1, F, F\_critical))

else:

print("θ{:d}: F = {:.2f} > {:.2f} \t=> Параметр значим".format(j + 1, F, F\_critical))

del temp

# Проверка о незначимости регрессии

print("Проверка гипотезы о незначимости регрессии с уровнем значимости {:.2f}".format(significance))

F\_critical = sp.f.isf(significance, len(theta\_vector)-1, len(X) - len(theta\_vector))

RSSh = sum((y - data["y"].mean())\*\*2 for y in data["y"].tolist())

RSS = (data["y"].to\_numpy() - X @ theta\_vector).T @ (data["y"].to\_numpy() - X @ theta\_vector)

F = ((RSSh-RSS) / (len(theta\_vector)-1)) / (RSS / (len(X) - len(theta\_vector)))

if F < F\_critical:

print("F = {:.2f} < {:.2f} \t=> Регрессия незначима".format(F, F\_critical))

else:

print("F = {:.2f} > {:.2f} \t=> Регрессия значима".format(F, F\_critical))

# Прогнозные значения

# Фиксируя x2

print("Прогноз математического ожидания с фиксированным x2 и уровнем доверия {:.2f}".format(1 - alpha))

x1\_values = [-1+2\*(j)/10 for j in range(11)]

x2\_values = np.ones(len(x1\_values))

def predict(x1\_values, x2\_values):

f\_array = np.array([[x1\_values[j] \*\* 2,

x1\_values[j],

x2\_values[j]\*\*3,

x2\_values[j]\*\*2,

1,

x1\_values[j]\*x2\_values[j]] for j in range(len(x1\_values))])

y = np.array([f\_x.T @ theta\_vector for f\_x in f\_array])

dev = np.array([abs(sp.t.ppf(alpha/2, len(X)-len(theta\_vector)) \*

np.sqrt(dispersion\_est)\*np.sqrt(f\_x.T @ np.linalg.inv(X.T @ X) @ f\_x)) for f\_x in f\_array])

return pd.DataFrame({

"x1": x1\_values,

"x2": x2\_values,

"left": y-dev,

"right": y+dev

})

prediction = predict(x1\_values, x2\_values)

print(prediction.round(2))

def plot\_prediction(prediction, column, file\_name):

plt.plot(prediction[column], prediction["left"], "-.")

plt.plot(prediction[column], prediction["right"], "-.")

plt.xlabel(column)

plt.ylabel("y")

plt.savefig(file\_name, bbox\_inches='tight')

plt.show()

plot\_prediction(prediction, "x1", "c:/Users/artem/Documents/Учёба/7 сем/СМАд/x1\_plot.png")

# Фиксируя x1

print("Прогноз математического ожидания с фиксированным x1 и уровнем доверия {:.2f}".format(1 - alpha))

x2\_values = [-1+2\*(j)/10 for j in range(11)]

x1\_values = np.ones(len(x2\_values))

prediction = predict(x1\_values, x2\_values)

print(prediction.round(2))

plot\_prediction(prediction, "x2", "c:/Users/artem/Documents/Учёба/7 сем/СМАд/x2\_plot.png")