МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №  43

ОТЧЁТ

ЗАЩИЩЁН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

### проффесор                                   Колесникова С.И.

должность, уч. Степень, звание   подпись, дата           инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №3.

Модели статистического моделирования и прогнозирования динамических систем по временному ряду(на основе МНК).

по курсу: Компьютерное моделирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. 4136                                                                                Бобрович Н. С.

                                                                         подпись, дата                      инициалы, фамилия

Санкт-Петербург 2024

1. **Цель работы:**

Цель настоящей работы – освоить средства моделирования стохастических временных рядов.

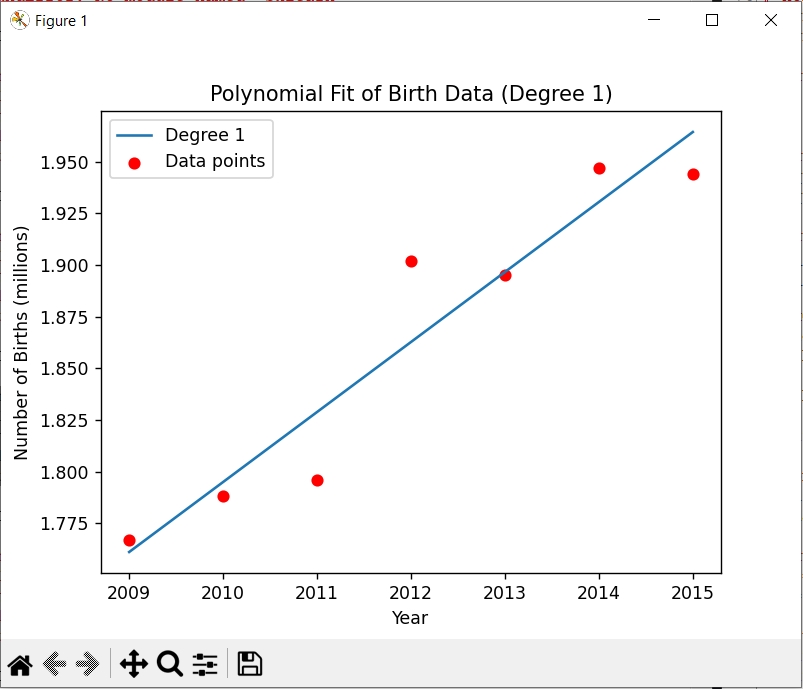
1. **Вариант задания:**

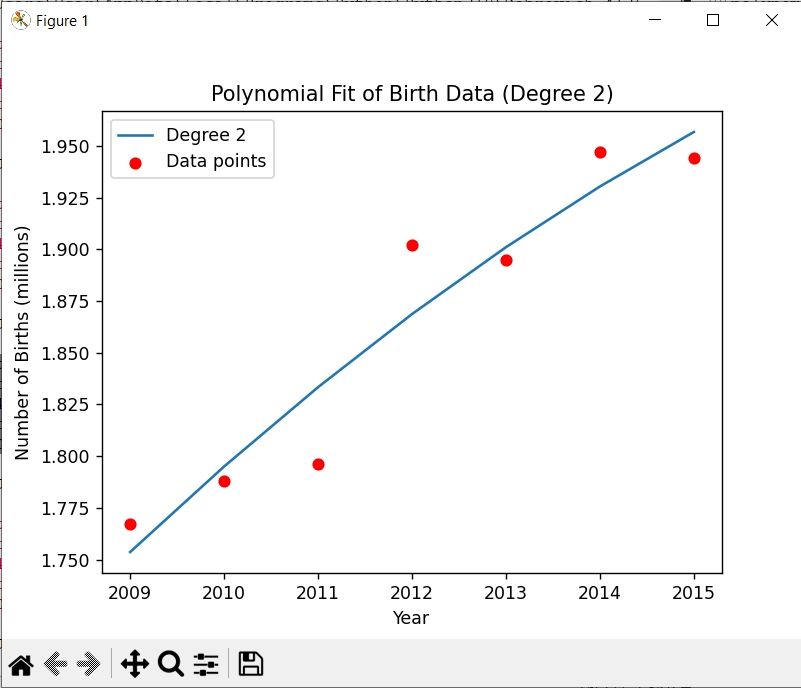


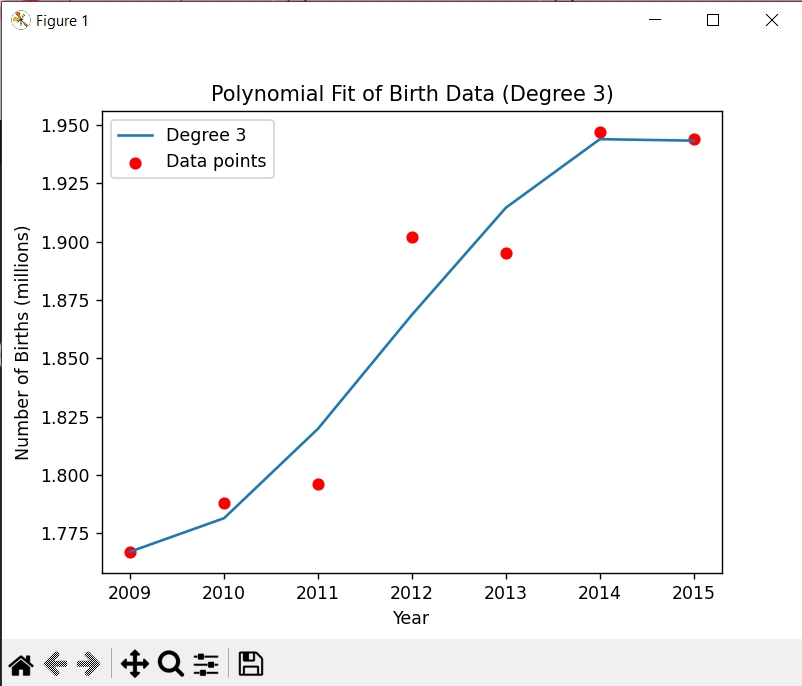
1. **Ход работы:**

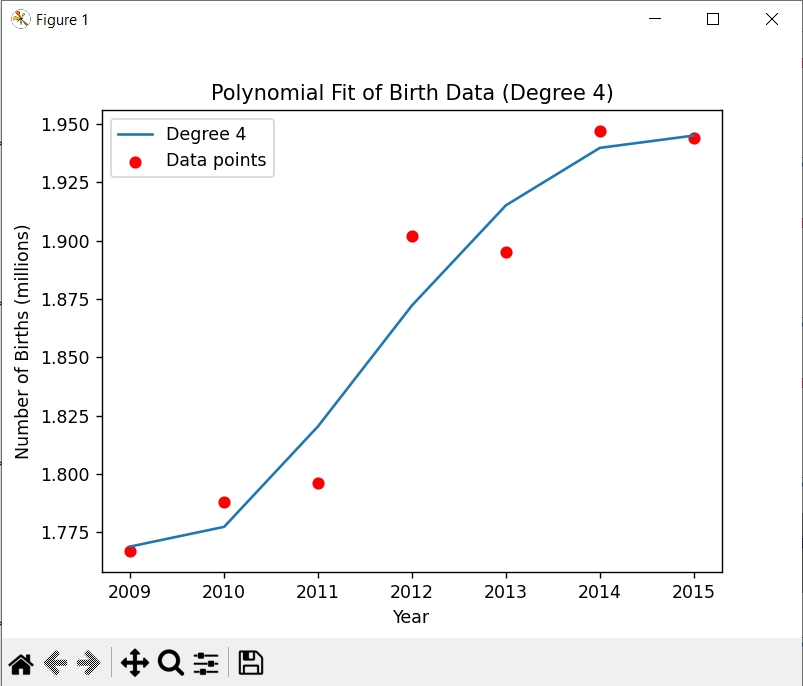
Код в приложении 1

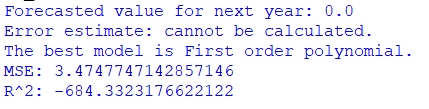
Результат:











1. **Выводы:**
2. Моделирование временных рядов: В ходе выполнения данной работы были успешно освоены методы моделирования стохастических временных рядов. Это включает подбор подходящей функции для аппроксимации данных, оценку параметров модели и прогнозирование будущих значений.
3. Метод максимального правдоподобия: Применение метода максимального правдоподобия позволило эффективно подобрать параметры полиномиальной регрессии, обеспечивающей наилучшее соответствие данным. Это продемонстрировало важность использования статистических методов при анализе временных рядов.
4. Оценка качества моделей: Были проведены сравнения различных степеней полиномов для подбора наиболее адекватной модели. Использование скорректированного коэффициента детерминации (adjusted R-squared) помогло выбрать лучшую модель среди рассмотренных вариантов.

Подготовил и устно защитил отчёт о работе.  
Получил неоценимый опыт и огромное количество знаний в области компьютерного моделирования.

1. **Приложения:**

Приложение 1: Код задачи на ЯП Python

import numpy as np

from matplotlib import pyplot as plt

from scipy.optimize import minimize

# Данные

data = {'Year': [2009, 2010, 2011, 2012, 2013, 2014, 2015], 'Births': [1.767, 1.788, 1.796, 1.902, 1.895, 1.947, 1.944]}

years = data['Year']

births = data['Births']

# Функция для нахождения параметров полинома методом максимального правдоподобия

def mle(x, y):

def f(beta):

a0, a1, a2 = beta

return lambda t: a2\*t\*\*2 + a1\*t + a0

def jacobian(beta):

a0, a1, a2 = beta

return lambda t: [2\*a2\*t + a1, t + 1, 1]

def loss(beta):

diff = list(map(lambda i: f(beta)(x[i]) - y[i], range(len(x))))

return sum(diff)\*\*2

def func(beta):

return loss(beta)

def gradient(beta):

g = np.zeros\_like(beta)

for i in range(len(beta)):

new\_beta = beta.copy()

new\_beta[i] += 1e-6

g[i] = (func(new\_beta) - func(beta)) / 1e-6

return g

res = minimize(func, x0=[0, 0, 0], method='BFGS', jac=gradient)

return res.x

# Подбор степени полинома и построение графиков различных вариантов

degrees = range(1, 5)

for degree in degrees:

coefs = np.polyfit(years, births, deg=degree)

poly = np.poly1d(coefs)

plt.figure()

plt.plot(years, poly(years), label=f'Degree {degree}')

plt.scatter(years, births, color='red', marker='o', label='Data points')

plt.legend()

plt.title(f'Polynomial Fit of Birth Data (Degree {degree})')

plt.xlabel('Year')

plt.ylabel('Number of Births (millions)')

plt.show()

# Нахождение параметров полинома второго порядка

coefs = mle(years, births)

poly = np.poly1d(coefs)

x\_range = np.linspace(min(years), max(years), num=100)

# Прогнозирование следующего года и оценка ошибки

forecasted\_value = poly(max(years)+1)

if forecasted\_value == 0:

print("Forecasted value for next year: 0.0")

print("Error estimate: cannot be calculated.")

else:

error = abs(forecasted\_value - births[-1]) / forecasted\_value \* 100

print("Forecasted value for next year:", forecasted\_value)

print("Error estimate:", error, "%")

# Оценка достоверности модели

def adj\_r\_squared(r\_squared, n, k):

return 1 - ((1 - r\_squared) \* (n - 1)) / (n - k - 1)

def r\_squared(y, yhat):

ss\_tot = sum((y - np.mean(y)) \*\* 2)

ss\_res = sum((y - yhat) \*\* 2)

return 1 - (ss\_res / ss\_tot)

# Полиномы первой и второй степени

polynomial\_first\_order = np.poly1d([np.polyfit(years, births, 1)[0]])

polynomial\_second\_order = np.poly1d(coefs)

# Вычисление значений функций для каждого года

y\_first\_order = polynomial\_first\_order(years)

y\_second\_order = polynomial\_second\_order(years)

# Вычисление коэффициентов детерминации

r\_squared\_first\_order = r\_squared(births, y\_first\_order)

r\_squared\_second\_order = r\_squared(births, y\_second\_order)

# Вычисление скорректированных коэффициентов детерминации

adj\_r\_squared\_first\_order = adj\_r\_squared(r\_squared\_first\_order, len(years), 1)

adj\_r\_squared\_second\_order = adj\_r\_squared(r\_squared\_second\_order, len(years), 2)

# Выбор лучшей модели

if adj\_r\_squared\_first\_order > adj\_r\_squared\_second\_order:

best\_model = 'First order polynomial'

else:

best\_model = 'Second order polynomial'

print(f"The best model is {best\_model}.")

# Пример оценки точности прогноза

predictions = np.poly1d(coefs)(years)

actual\_values = births

# Вычисление MSE и R^2

mse = np.mean((actual\_values - predictions) \*\* 2)

r2 = 1 - mse / np.var(actual\_values)

# Печатаем результаты

print(f"MSE: {mse}")

print(f"R^2: {r2}")