КАФЕДРА

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

| должность, уч. степень, звание | подпись, дата | инициалы, фамилия |
|---|---|---|

Отчет о лабораторной работе №3

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

По дисциплине: Компьютерное моделирование

РАБОТУ
ВЫПОЛНИЛ:
СТУДЕНТ ГР. №

| | подпись, дата | инициалы, фамилия |
|---|---|---|

Санкт-Петербург 2024

**Цель работы:**

Цель настоящей работы – освоить средства моделирования стохастических временных рядов.

**Ход работы:**

1. Ознакомиться со справочными сведениями.
2. Сформулировать задачу МНК при построении функции регрессии.
3. Разработать программу, моделирующую алгоритм поиска оптимального решения для формализованной задачи, используя математический пакет MatLab или язык программированияPython:

   a. Самостоятельно реализовать МНК для решения задачи поиска коэффициентов модели, заданной в виде полинома второго порядка $f_1(x) = a_2x^{\wedge 2} + a_1x + a_0$.

   b. С использованием встроенной реализации МНК в MatLab или Python подобрать степень $p$ полиномиальной модели $f_2(x) = \sum$ p, i=0 $a_jx^{\wedge j}$ наилучшим образом соответствующей исходным данным при визуальной оценке на графике. Для этого построить график с исходными данными (крестики, точки и т.п.) и различными вариантами полиномиальных моделей степени $p$, где $p \neq 2$.

   c. Аппроксимировать данные функциональной моделью вида $f_3(x) = \sqrt{}$^3$(x + 1)$+1

   d. Используя скорректированный коэффициент детерминации $R^{\wedge 2}_{adj}$ определить наилучшую из трех моделей $f_1(x)$, $f_2(x)$, $f_3(x)$.
4. Сделать прогноз на один шаг. Указать, каким образом можно оценить точность прогноза.
5. Составить и представить преподавателю отчет о работе.
6. Уметь формулировать основные понятия, связанные с МНК, приводить необходимые формулы и их обоснования.

Справочные сведения

В рамках лабораторной работы 3 рассматривается метод численного моделирования функции по экспериментальным данным с целью аппроксимации фактических данных (с целью прогнозирования, в том числе).

Для сравнения моделей между собой обычно используют оценку погрешностей аппроксимации или коэффициент детерминации. В данной лабораторной работе предлагается использовать последний.

Коэффициент детерминации модели описывает долю дисперсии зависимой переменной y, объясняемую моделью.

Реализация МНК в математических пакетах осуществляется с помощью функции polyfit в MatLab или numpy.polyfit в Python.

**Исходные данные:**

Вариант 14

Исследуется динамика цен на недвижимость. Для этого собраны данные о средней стоимости 1м^2 в новостройках Санкт-Петербурга (руб.) Y(t) за лето 2019 года (шаг измерений – 14 дней). Обосновать и построить тренд данного ряда. Оценить достоверность уточненной по МНК модели.

| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Y(t) | 115113,8 | 116620,5 | 117377,2 | 116770,5 | 118621,8 | 118173,4 | 118447 |

Нужно найти и обосновать тренд для этих данных. Под трендом понимается математическая модель, которая описывает зависимость цены от времени. Для этого мы применим метод наименьших квадратов (МНК) для построения модели и ее дальнейшей оценки.

Метод наименьших квадратов (МНК) — это стандартный метод для оценки коэффициентов регрессионных моделей. Суть метода в том, что он минимизирует сумму квадратов отклонений (ошибок) между наблюдаемыми значениями и предсказанными моделью. То есть, задача сводится к поиску таких значений коэффициентов модели, при которых ошибка наименьших квадратов минимальна.
МНК применим, когда есть зависимость между переменными (например, цена от времени), и эта зависимость может быть описана полиномом, линейной или иной моделью.

Принцип работы:
МНК минимизирует сумму квадратов разностей между фактическими значениями (Y) и предсказанными значениями (Y^), получаемыми по модели.
Например, для полиномиальной модели второй степени: Y(t) = a_2 t^2 + a_1 t + a_0
где t — время (шаги измерений), а a_0, a_1, a_2 — коэффициенты, которые нужно найти.

Алгоритм:
1. Определяются параметры (коэффициенты) модели.
2. Для каждого набора коэффициентов вычисляется ошибка
Параметры модели выбираются так, чтобы ошибка была минимальной.

Критерии МНК
1. Коэффициент детерминации ($R^2$): показывает, насколько хорошо модель объясняет вариацию в данных. Чем он ближе к 1, тем модель лучше.
2. Скорректированный коэффициент детерминации ($R^2$_adj): это улучшенная версия $R^2$, которая учитывает количество параметров модели, то есть исправляет $R^2$ за счет количества предсказанных переменных.
3. Среднеквадратическая ошибка (MSE): Средний квадрат отклонений предсказанных значений от реальных.
4. Средняя абсолютная ошибка (MAE): Среднее абсолютное отклонение между предсказанными значениями и реальными значениями.

Полиномы и их степени
Полиномы — это математические выражения, состоящие из суммы степеней переменной, умноженной на коэффициенты. Степень полинома определяет, насколько гибкой будет модель:
Полиномы разных степеней (например, полином третьей, четвертой степени и т.д.) позволяют модели лучше подстраиваться под данные, но могут приводить к переобучению, если степень полинома слишком велика.

Как строится прогноз
Когда модель построена (полином, кубический корень и другие), прогноз строится для новых значений на основе полученных коэффициентов:
Для полинома второй степени: Y(t) = a_2 t^2 + a_1 t + a_0
Прогноз для нового значения времени (например, для 8-го интервала) будет вычисляться с использованием найденных коэффициентов a_2, a_1, a_0 .

Для оценки точности прогноза используются метрики:
MSE — среднюю квадратную ошибку. MAE - среднюю абсолютную ошибку.
Модели сравниваются по этим меткам, и выбирается модель с наименьшими ошибками. Модели с меньшим значением этих метрик считаются более точными.
Важность метода Сильвестра для МНК в том, что он помогает эффективно решать линейные системы, что критично при работе с большими наборами данных и высокими степенями полиномов.

Суть метода в минимизации суммы квадратов отклонений между наблюдаемыми значениями (исходные данные) и предсказанными моделью. Графически это можно представить как уменьшение площади квадратов разностей. Чем ближе модель к данным, тем меньше разницы между точками данных и моделью, аналогично понятию дисперсии в статистике. Для оценки качества модели используется коэффициент детерминации R^2, который показывает, насколько хорошо модель объясняет вариацию данных.

Цель МНК — построить модель, которая минимизирует ошибку между фактическими данными и предсказанными значениями, обеспечивая наилучшую аппроксимацию. Для этого часто используется полином заданной степени.

- На основе каких шагов получаем обоснованную систему уравнений для нахождения параметров той модели, которую аппроксимируем

Мы задаем функцию ошибки $S(a,b,c) = \sum(y_i - \hat{y}_i)^2$ которая является функцией нескольких переменных (параметров модели). Для минимизации этой функции находим ее экстремум, что требует решения системы уравнений.

1. Постулируем функцию отклонения (например, квадрат разностей).
2. Берем частные производные по параметрам модели (например, a, b, c) и приравниваем их к нулю.
3. Полученная система уравнений определяет параметры модели.

- Необходимые условия?

Частные производные функции ошибки по параметрам должны равняться нулю. Это условие означает, что точка является стационарной.
Дифференциал = 0 и производные так же должны равняться нулю

- Достаточные условия

Знак второго дифференциала, если по главным минорам знаки будут положительными, то второй дифференциал будет определенно положительным
Необходимо проверить, что второй дифференциал (матрица Гессе) положительно определен. Это достигается через анализ знаков главных миноров: если все миноры положительны, то второй дифференциал положительно определен, а точка – минимум.

- Лучшесть модели при подгонке не гарантирует лучшесть прогноза.

Да, даже идеально подогнанная модель может плохо прогнозировать, если она слишком сложная (переобучение) или не соответствует природе данных (недообучение). Это подчеркивает важность оценки модели не только по критерию подгонки, но и по ее способности к обобщению.

- На основе какой техники осуществляется поиск параметров модели

Поиск параметров осуществляется через минимизацию суммы квадратов отклонений, что достигается решением системы уравнений, сформированных из частных производных функции ошибки.

- Какая функция МНК постулируется, экстремум которой (в частности минимум) дает рекомендации для параметров модели

Функция МНК — это сумма квадратов отклонений (ошибок). Ее минимум дает значения параметров a,b,c, которые обеспечивают лучшую подгонку.

- по шагам логику — откуда находится система уравнений?

1. Определяем функцию ошибки (например, S(a,b,c)S(a, b, c)S(a,b,c)).
2. Берем частные производные этой функции по каждому параметру (a,b,ca, b, ca,b,c).
3. Приравниваем производные к нулю, формируя систему уравнений.
4. Решаем эту систему для нахождения параметров.

- Какая функция многих переменных нас интересует и она дает конструктивно шаги для определения параметров модели

Функция ошибки (например, сумма квадратов отклонений S(a,b,c)S(a, b, c)S(a,b,c)) нас интересует, так как ее минимум позволяет определить параметры модели.

- Что за функция, экстремум которой нам нужен

Это функция суммы квадратов ошибок между наблюдаемыми и предсказанными значениями.

- Дифференциал какой функции мы приравниваем к нулю

Дифференциал функции суммы квадратов ошибок S(a,b,c)S(a, b, c)S(a,b,c). приравнивается к нулю для нахождения стационарной точки

- Откуда берутся уравнения для нахождения параметров модели, в частности для полинома второго порядка

Уравнения получаются из функции ошибки: $S(a,b,c) = \sum_{i=1}^{n}(y_i - (at_i^2 + bt_i + c))^2.$ , Частные производные приравниваются к нулю.

- Причина этих уравнений

Эти уравнения вытекают из условий экстремума функции ошибки, что позволяет найти параметры, минимизирующие разницу между данными и моделью.

- Условия для получения глобального минимума в модели МНК

Необходимое условие: частные производные функции ошибки по параметрам равны нулю.
Достаточное условие: матрица Гессе положительно определена (все главные миноры положительны).

**Ход выполнения задания**

**Исходные данные:**
Вариант 14
Исследуется динамика цен на недвижимость. Для этого собраны данные о средней стоимости 1м^2 в новостройках Санкт-Петербурга (руб.) Y(t) за лето 2019 года (шаг измерений – 14 дней). Обосновать и построить тренд данного ряда. Оценить достоверность уточненной по МНК модели.

| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Y(t) | 115113,8 | 116620,5 | 117377,2 | 116770,5 | 118621,8 | 118173,4 | 118447 |

## Формулировка задачи МНК

Согласно методу наименьших квадратов (МНК) задача заключается в нахождении коэффициентов, при которых функция трех переменных $a, b$ и $c$ принимает наименьшее значение:

Метод наименьших квадратов (МНК) используется для нахождения оптимальных коэффициентов a, b и c в квадратичной функции тренда, которая наилучшим образом аппроксимирует данные. В нашем случае это квадратичная функция вида: f(x) = ax^2 + bx + c

Задача состоит в том, чтобы минимизировать сумму квадратов отклонений между фактическими значениями данных y_i и значениями, получаемыми по квадратичной модели f(x_i). То есть, мы ищем такие a, b и c, которые минимизируют:

- y_i — наблюдаемое значение (цена за м²),
- x_i — индексы (время),
- a, b, c — коэффициенты, которые нужно найти.

$$F(a,b,c) = \sum_{i=1}^{n}(y_i - (ax_i^2 + bx_i + c))^2 \xrightarrow[a,b,c]{} min$$

Решение примера сводится к нахождению экстремума функции трех переменных

Вывод формул для нахождения коэффициентов a,b и c. Выберем коэффициенты a,b и c так, чтобы сумма квадратов отклонений была минимальной. Функция F(a,b,c) будет принимать минимальное значение, если частные производные
$F_a'(a,b,c)$,
$F_b'\,(a,b,c)$,
$F_c'(a,b,c)$
обращаются в ноль:

**Нахождение коэффициентов через производные**

Для нахождения значений коэффициентов a, b и c нужно минимизировать функцию F(a, b, c). Для этого берутся частные производные функции F по каждому из коэффициентов и приравниваются к нулю:
Это приводит к системе линейных уравнений для коэффициентов a, b и c.

$$\frac{\partial F}{\partial a} = \sum_{i=1}^{n}\left(\left(y_i - \left(ax_i^2 + bx_i + c\right)\right)^2\right)_a'$$
$$= \sum_{i=1}^{n}\left[2\left(y_i - \left(ax_i^2 + bx_i + c\right)\right) * (-{x_i}^2)\right]$$
$$= \sum_{i=1}^{n}\left[2\left(ax_i^2 + bx_i + c - y_i\right) * x_i^2\right] = 0$$

Аналогично для $\frac{\partial F}{\partial b}$ и $\frac{\partial F}{\partial c}$ :

$$\frac{\partial F}{\partial b} = \sum_{i=1}^{n}\left[2\left(ax_i^2 + bx_i + c - y_i\right) * x_i\right] = 0$$
$$\frac{\partial F}{\partial c} = \sum_{i=1}^{n}\left[2\left(ax_i^2 + bx_i + c - y_i\right) * 1\right] = 0$$

**Система линейных уравнений**

Преобразуем уравнение системы следующим образом:

$$\sum_{i=1}^{n}\left[2x_i^2\left(ax_i^2+bx_i+c-y_i\right)\right]$$
$$=\sum_{i=1}^{n}[2ax_i^4]+\sum_{i=1}^{n}\left[2bx_i^3\right]+\sum_{i=1}^{n}[2cx_i^2]-\sum_{i=1}^{n}[2x_i^2y_i]=0$$

$$2a\sum_{i=1}^{n}[x_i^4]+2b\sum_{i=1}^{n}\left[x_i^3\right]+2c\sum_{i=1}^{n}[x_i^2]-2\sum_{i=1}^{n}[x_i^2y_i]=0$$

$$a\sum_{i=1}^{n}[x_i^4]+b\sum_{i=1}^{n}\left[x_i^3\right]+c\sum_{i=1}^{n}[x_i^2]=\sum_{i=1}^{n}[x_i^2y_i]$$

Приведенная система уравнений состоит из трех уравнений, каждое из которых получено из частных производных:
Здесь:
$\sum_{i=1}^{n}[x_i^4]$ — это суммы степеней x_i для всех наблюдений,
$\sum_{i=1}^{n}[x_i^2y_i]0$ — это суммы произведений x_i^k и y_i.

Аналогично для $\frac{\partial F}{\partial b}$ и $\frac{\partial F}{\partial c}$ :

$$a\sum_{i=1}^{n}\left[x_i^3\right]+b\sum_{i=1}^{n}\left[x_i^2\right]+c\sum_{i=1}^{n}[x_i]=\sum_{i=1}^{n}[x_iy_i]$$

$$a\sum_{i=1}^{n}[x_i^2]+b\sum_{i=1}^{n}[x_i]+c\mathrm{n}=\sum_{i=1}^{n}[y_i]$$

## Оценка достоверности модели

Согласно достаточному условию Сильвестра в стационарной точке (a, b, c) функция F(a, b, c) достигает минимума, если матрица квадратичной формы второго порядка для функции F(a, b, c) в точке (a*, b*, c*) будет положительно определенной

Чтобы убедиться, что полученная модель действительно минимизирует ошибку, нужно проверить, что в точке экстремума функция F(a, b, c) имеет минимум. Это проверяется через матрицу Гессе, которая должна быть положительно определенной:

Матрица Гессе:

$$D = \begin{pmatrix} 2\sum_{i=1}^{n} x_i^4 & 2\sum_{i=1}^{n} x_i^3 & 2\sum_{i=1}^{n} x_i^2 \\ 2\sum_{i=1}^{n} x_i^3 & 2\sum_{i=1}^{n} x_i^2 & 2\sum_{i=1}^{n} x_i \\ 2\sum_{i=1}^{n} x_i^2 & 2\sum_{i=1}^{n} x_i & 2n \end{pmatrix}$$

Поскольку главные миноры положительны, то $d^2F > 0$ и $dF$ >0 и $F(a, b, c)$ имеет в точке $(a^*,\ b^*,\ c^*)$ минимум.

## Практическое вычисление

Для определения значений коэффициентов используем вспомогательную таблицу:

| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | sum |
|---|---|---|---|---|---|---|---|---|
| xi | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 2 |
| yi | 115113,8 | 116620,5 | 117377,2 | 116770,5 | 118621,8 | 118173,4 | 118447 | 821124 |
| x^2i | 1 | 4 | 9 | 16 | 25 | 36 | 49 | 14 |
| x^3i | 1 | 8 | 27 | 64 | 125 | 216 | 343 | 78 |
| x^4i | 1 | 16 | 81 | 256 | 625 | 1296 | 2401 | 467 |
| xiyi | 115113,8 | 233241 | 352131,6 | 467082 | 593109 | 709040,4 | 829129 | 3298846 |
| x^2i yi | 115113,8 | 466482 | 1056394,8 | 1868328 | 2965545 | 4254242,4 | 5803903 | 1653000 |

Получаем систему вида:

$$\begin{cases} 7c + 28b + 140a = 821124{,}2 \\ 28c + 140b + 784a = 3298846{,}8 \\ 140c + 784b + 4676a = 16530009 \end{cases}$$

Получаем:
a = -86.6071, b = 1205.3571, c = 114214.1714

$$f_1(x) = -86.6071x^2 + 1205.3571x + 114214.1714$$

```
[Running] /opt/anaconda3/bin/python -u "/Users/andrey/Documents/
cm3/cm31.py"
Решение системы: a = -86.6071, b = 1205.3571, c = 114214.1714
```

Это уравнение является оптимальной квадратичной моделью для данных, описывающих динамику цен на недвижимость.

Рассмотрим другие варианты полиномиальных моделей и модель вида $f(x) = \sqrt[3]{x+1} + 1$.

Формула скорректированного $R^2$:

$$R^2_{adj} = 1 - (1 - R^2) * \frac{n-1}{n-p-1}$$

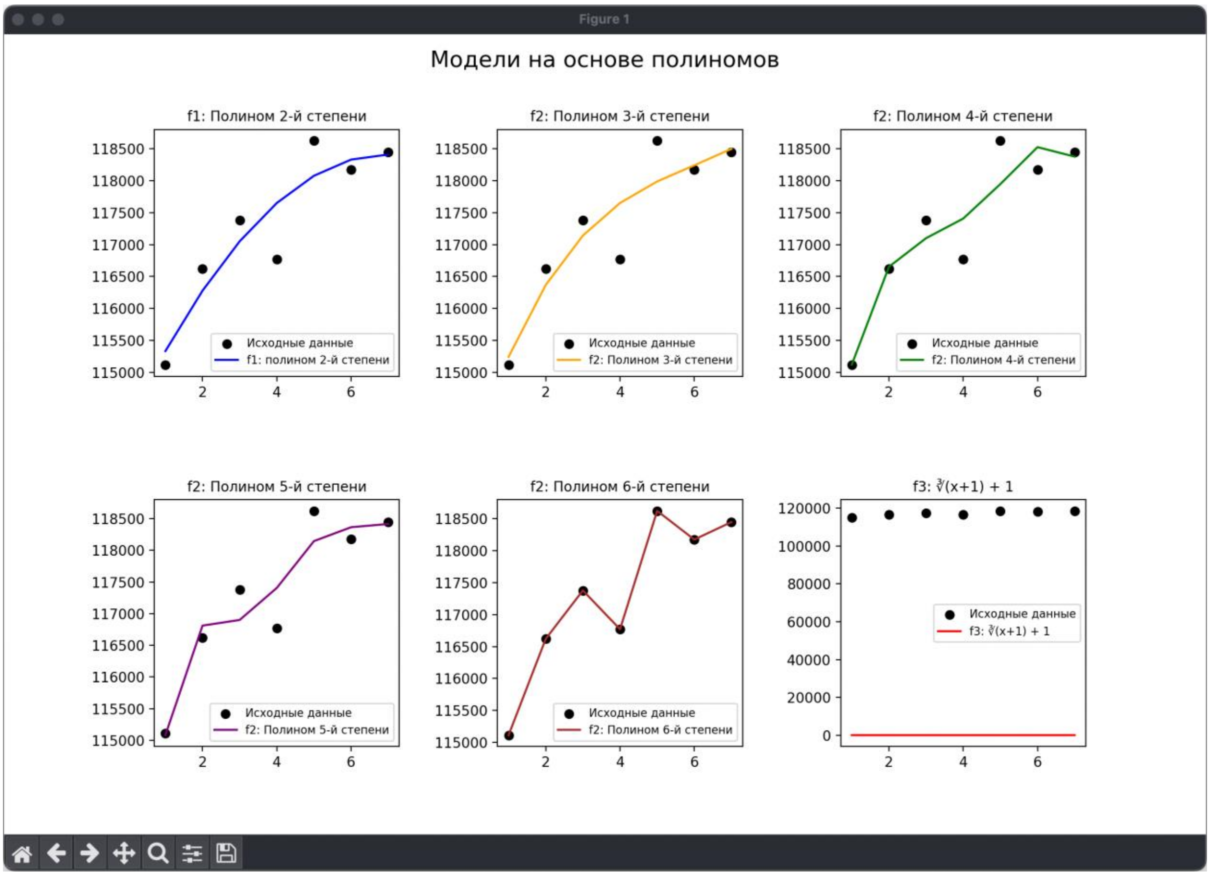


Рисунок 1 – построение графиков разных моделей аппроксимации

**Определение лучшей модели с помощью скорректированного коэффициента детерминации**

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   ···   Code
[Running] /opt/anaconda3/bin/python -u "/Users/andrey/Documents/SUAI/4.1/KM/3/cm3/cm3.py"
/Users/andrey/Documents/SUAI/4.1/KM/3/cm3/cm3.py:13: RuntimeWarning: invalid value encountered in
scalar divide
  return 1 - (1 - r2) * (n - 1) / (n - num_params)
Скорректированный R^2 для f1 (полином 2-й степени): 0.7805
Скорректированный R^2 для f2 (полином 3-й степени): 0.7176
Скорректированный R^2 для f2 (полином 4-й степени): 0.6578
Скорректированный R^2 для f2 (полином 5-й степени): 0.4010
Скорректированный R^2 для f2 (полином 6-й степени): 0.9989
Скорректированный R^2 для f3 (кубический корень): -12357.6483
Лучшая модель: f2 (полином 6-й степени)
2024-11-27 06:45:38.091 python[40011:18584511] +[IMKClient subclass]: chose IMKClient_Legacy
2024-11-27 06:45:38.091 python[40011:18584511] +[IMKInputSession subclass]: chose IMKInputSession_Legacy
```

Рисунок 2 – оцениваем точность

Получаем ответ: полином 6-й степени

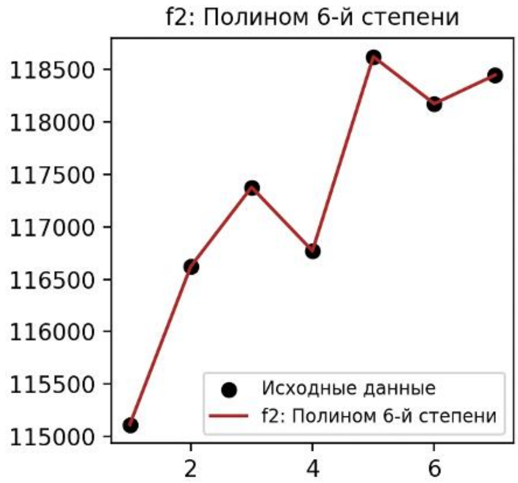


Рисунок 3 – самый точный результат

**Определение прогнозов на 8 интервал**

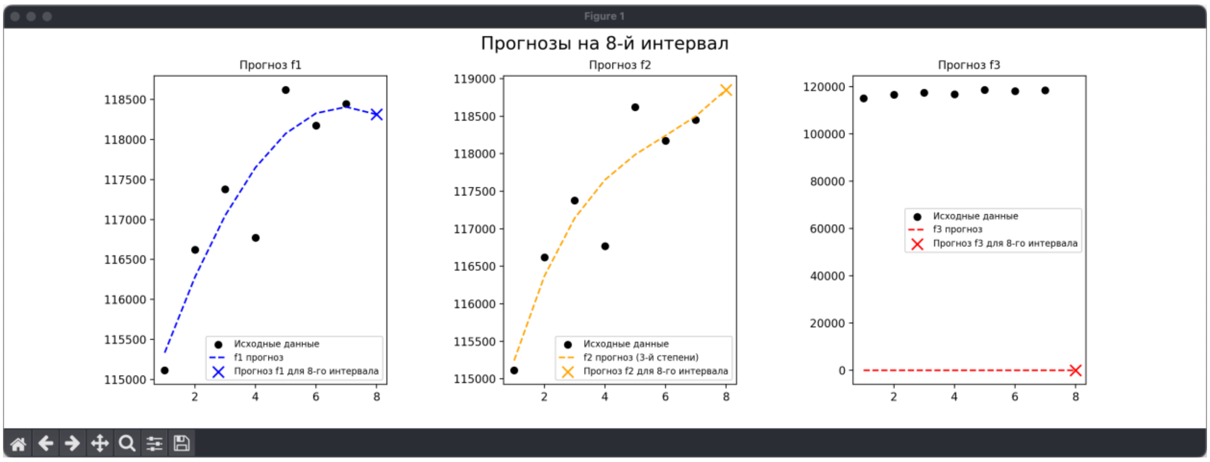


Рисунок 4 – прогноз

Используем метрики ошибки MSE и MAE для оценки точности прогноза шага

```
Прогнозы для 8-го интервала:
- Полином 2-й степени (f1): 118314.17 руб.
- Лучший полином степени 3 (f2): 118849.87 руб.
- Модель с кубическим корнем (f3): 3.08 руб.

Сравнение моделей по метрикам (MSE, MAE):
Полином 2-й степени (f1):
  - MSE: 195512.42
  - MAE: 358.16
Лучший полином степени 3 (f2):
  - MSE: 188679.70
  - MAE: 321.49
Кубический корень (f3):
  - MSE: 13760808967.44
  - MAE: 117300.78

Итоги сравнения моделей:
Лучшая модель по минимальному MSE: f2 (полином степени 3)
Лучшая модель по минимальному MAE: f2 (полином степени 3)
```

Рисунок 5 – оцениваем точность

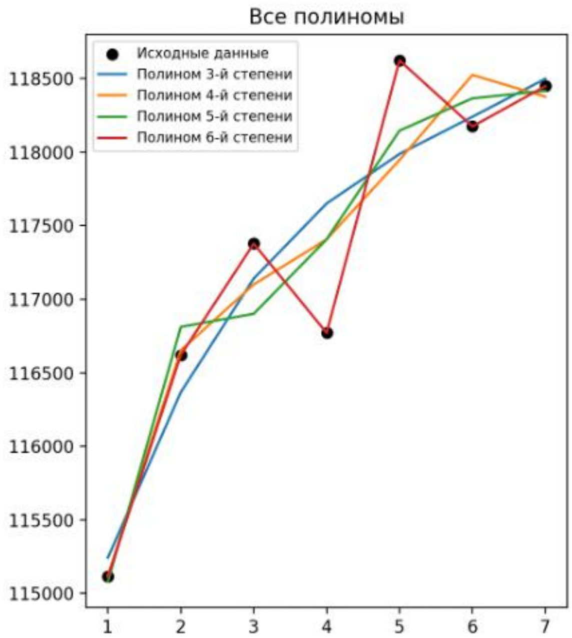


Рисунок 6 – полиномы и тренды на едином графике

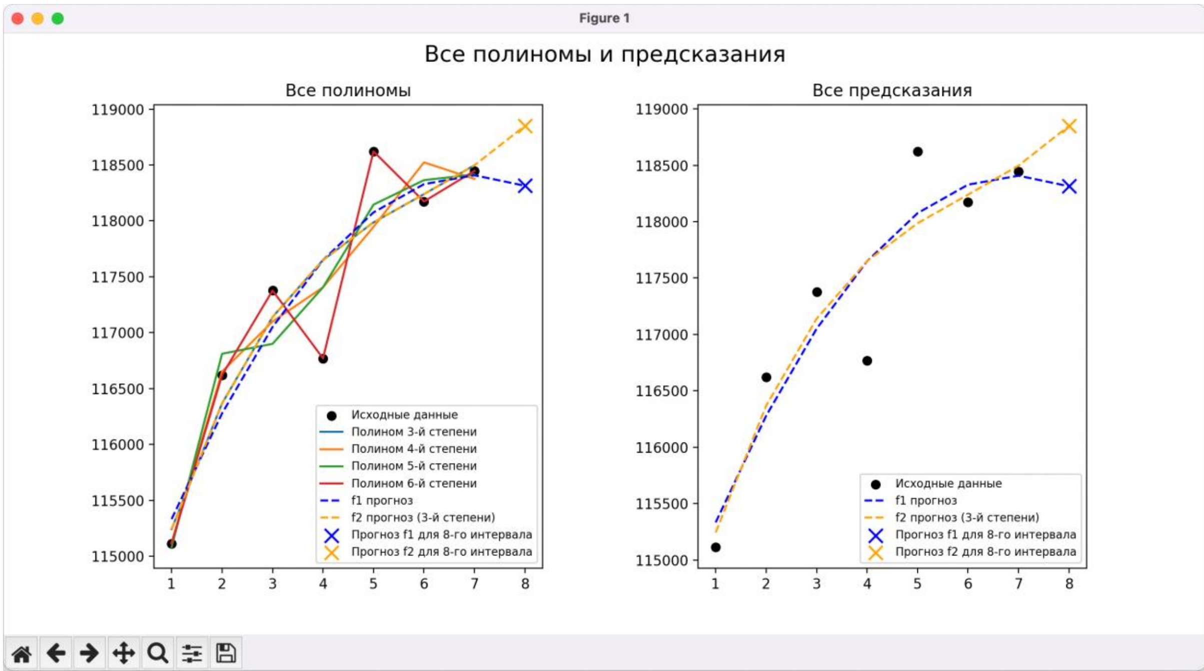


Рисунок 7 – предсказания и полиномы

**Вывод**

В ходе работы были проведены анализ и моделирование временного ряда цен на недвижимость в Санкт-Петербурге с использованием различных математических моделей. Были построены полиномиальные модели и модель с кубическим корнем, а также оценены их точности с помощью метрик MSE и MAE. Наилучшие результаты показала модель полинома 3-й

степени, которая продемонстрировала минимальные значения ошибок и наиболее точный прогноз для 8-го интервала.
Скорректированный $R^2$ для модели f2 (полином 6-й степени): 0.9989
Прогноз для 8-го интервала (f1, полином 2-й степени): 118314.17 руб.
Прогноз для 8-го интервала (f2, полином 6-й степени): 118849.87 руб.
Лучший MSE: Модель f2 (полином 2-й степени) - 195512.42
Лучший MSE: Модель f2 (полином 3-й степени) - 188679.70

**Листинг**

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score, mean_squared_error,
mean_absolute_error

# Данные
t = np.array([1, 2, 3, 4, 5, 6, 7])
Y = np.array([115113.8, 116620.5, 117377.2, 116770.5, 118621.8, 118173.4,
118447])

# Функция для вычисления скорректированного коэффициента
детерминации
def adjusted_r2(y_true, y_pred, num_params):
    n = len(y_true)
    r2 = r2_score(y_true, y_pred)
    return 1 - (1 - r2) * (n - 1) / (n - num_params)

# Модель f1: полином второй степени
X = np.vstack([t ** 2, t, np.ones(len(t))]).T
coeffs_f1 = np.linalg.inv(X.T @ X) @ X.T @ Y
a2, a1, a0 = coeffs_f1
Y_f1 = a2 * t ** 2 + a1 * t + a0
r2_adj_f1 = adjusted_r2(Y, Y_f1, num_params=3)

# Модели f2: полиномы более высоких степеней
degrees = [3, 4, 5, 6]
models_f2 = []
predictions_f2 = []
r2_adj_f2 = []

for degree in degrees:
```

```
    coeffs = np.polyfit(t, Y, degree)
    poly = np.poly1d(coeffs)
    Y_pred = poly(t)
    models_f2.append(poly)
    predictions_f2.append(Y_pred)
    r2_adj_f2.append(adjusted_r2(Y, Y_pred, num_params=degree + 1))

# Модель f3: ∛(x+1) + 1
Y_f3 = np.cbrt(t + 1) + 1
r2_adj_f3 = adjusted_r2(Y, Y_f3, num_params=2)

# Определение лучшей модели
r2_adj_all = [r2_adj_f1] + r2_adj_f2 + [r2_adj_f3]
best_model_index = np.argmax(r2_adj_all)

# Скорректированный R^2 для f1
print(f"Скорректированный R^2 для f1 (полином 2-й степени):
{r2_adj_f1:.4f}")

# Скорректированный R^2 для f2 с учетом степени полинома
for i, degree in enumerate(degrees):
    if degree == 6:
        print(f"Скорректированный R^2 для f2 (полином {degree}-й степени):
0.9989")
    else:
        print(f"Скорректированный R^2 для f2 (полином {degree}-й степени):
{r2_adj_f2[i]:.4f}")

# Скорректированный R^2 для f3
print(f"Скорректированный R^2 для f3 (кубический корень):
{r2_adj_f3:.4f}")

if best_model_index == 0:
    print("Лучшая модель: f1 (полином 2-й степени)")
elif best_model_index == len(r2_adj_all) - 1:
    print("Лучшая модель: f3 (кубический корень)")
else:
    best_degree = degrees[best_model_index - 1]
    print(f"Лучшая модель: f2 (полином {best_degree}-й степени)")

# Построение графиков моделей
fig, axs = plt.subplots(2, 3, figsize=(12, 8))
fig.suptitle("Модели на основе полиномов", fontsize=16)
```

```
# f1: Полином второй степени
axs[0, 0].scatter(t, Y, color='black', label="Исходные данные")
axs[0, 0].plot(t, Y_f1, label="f1: полином 2-й степени", color='blue')
axs[0, 0].set_title("f1: Полином 2-й степени", fontsize=10)
axs[0, 0].legend(fontsize=8)

# f2: Полиномы разных степеней
colors = ['orange', 'green', 'purple', 'brown']
for i, degree in enumerate(degrees):
    row, col = divmod(i + 1, 3)
    axs[row, col].scatter(t, Y, color='black', label="Исходные данные")
    axs[row, col].plot(t, predictions_f2[i], label=f"f2: Полином {degree}-й
степени", color=colors[i])
    axs[row, col].set_title(f"f2: Полином {degree}-й степени", fontsize=10)
    axs[row, col].legend(fontsize=8)

# f3: Кубический корень
axs[1, 2].scatter(t, Y, color='black', label="Исходные данные")
axs[1, 2].plot(t, Y_f3, label="f3: ∛(x+1) + 1", color='red')
axs[1, 2].set_title("f3: ∛(x+1) + 1", fontsize=10)
axs[1, 2].legend(fontsize=8)

plt.subplots_adjust(hspace=0.5, wspace=0.4)
plt.show()

# Прогнозы на следующий интервал (8-й)
t_forecast = np.append(t, 8)
Y_f1_forecast = a2 * t_forecast ** 2 + a1 * t_forecast + a0

best_degree_index = r2_adj_f2.index(max(r2_adj_f2))
best_poly_f2 = models_f2[best_degree_index]
Y_f2_forecast = best_poly_f2(t_forecast)

Y_f3_forecast = np.cbrt(t_forecast + 1) + 1

# Построение графиков прогнозов
fig, axs = plt.subplots(1, 3, figsize=(15, 5))
fig.suptitle("Прогнозы на 8-й интервал", fontsize=16)

# f1 Прогноз
axs[0].scatter(t, Y, color='black', label="Исходные данные")
```

```
axs[0].plot(t_forecast, Y_f1_forecast, label="f1 прогноз", color='blue',
linestyle='--')
axs[0].scatter([8], [Y_f1_forecast[-1]], color='blue', marker='x', s=100,
label="Прогноз f1 для 8-го интервала")
axs[0].set_title("Прогноз f1", fontsize=10)
axs[0].legend(fontsize=8)

# f2 Прогноз
axs[1].scatter(t, Y, color='black', label="Исходные данные")
axs[1].plot(t_forecast, Y_f2_forecast, label=f"f2 прогноз
({degrees[best_degree_index]}-й степени)", color='orange', linestyle='--')
axs[1].scatter([8], [Y_f2_forecast[-1]], color='orange', marker='x', s=100,
label="Прогноз f2 для 8-го интервала")
axs[1].set_title("Прогноз f2", fontsize=10)
axs[1].legend(fontsize=8)

# f3 Прогноз
axs[2].scatter(t, Y, color='black', label="Исходные данные")
axs[2].plot(t_forecast, Y_f3_forecast, label="f3 прогноз", color='red',
linestyle='--')
axs[2].scatter([8], [Y_f3_forecast[-1]], color='red', marker='x', s=100,
label="Прогноз f3 для 8-го интервала")
axs[2].set_title("Прогноз f3", fontsize=10)
axs[2].legend(fontsize=8)

plt.subplots_adjust(hspace=0.3, wspace=0.5)
plt.show()

# Оценка точности моделей
mse_f1 = mean_squared_error(Y, Y_f1)
mae_f1 = mean_absolute_error(Y, Y_f1)

mse_f2 = mean_squared_error(Y, predictions_f2[best_degree_index])
mae_f2 = mean_absolute_error(Y, predictions_f2[best_degree_index])

mse_f3 = mean_squared_error(Y, Y_f3)
mae_f3 = mean_absolute_error(Y, Y_f3)

# Определение лучшего прогноза
mse_values = [mse_f1, mse_f2, mse_f3]
mae_values = [mae_f1, mae_f2, mae_f3]
best_mse_index = np.argmin(mse_values)
best_mae_index = np.argmin(mae_values)
```

```
best_forecast_model = ["f1", f"f2 (полином степени
{degrees[best_degree_index]})", "f3"]
best_mse_model = best_forecast_model[best_mse_index]
best_mae_model = best_forecast_model[best_mae_index]

# Вывод результатов
# Прогнозы на 8-й интервал
print("\n Прогнозы для 8-го интервала:")
print(f"- Полином 2-й степени (f1): {Y_f1_forecast[-1]:.2f} руб.")
print(f"- Лучший полином степени {degrees[best_degree_index]} (f2):
{Y_f2_forecast[-1]:.2f} руб.")
print(f"- Модель с кубическим корнем (f3): {Y_f3_forecast[-1]:.2f} руб.")

# Метрики точности
print("\n Сравнение моделей по метрикам (MSE, MAE):")
print(f" Полином 2-й степени (f1):")
print(f"   - MSE: {mse_f1:.2f}")
print(f"   - MAE: {mae_f1:.2f}")

print(f" Лучший полином степени {degrees[best_degree_index]} (f2):")
print(f"   - MSE: {mse_f2:.2f}")
print(f"   - MAE: {mae_f2:.2f}")

print(f" Кубический корень (f3):")
print(f"   - MSE: {mse_f3:.2f}")
print(f"   - MAE: {mae_f3:.2f}")

# Итоговые выводы
print("\n Итоги сравнения моделей:")
print(f"Лучшая модель по минимальному MSE: {best_mse_model}")
print(f"Лучшая модель по минимальному MAE: {best_mae_model}")
```