ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Дожность		Колесникова С.И
старший преподаватель	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.	4236		Л. Мвале
		подпись, дата	инициалы, фамилия

Санкт-Петербург
2025

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

$$f_1(x) = a_2x^2 + a_1x + a_0.$$

   b. С использованием встроенной реализации МНК в MatLab илиPython подобрать степень p полиномиальной модели $f_2(x) = \sum_{i=0}^{p} a_i x^i,$, 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 ≠ 2.
   c. Аппроксимировать данные функциональной моделью вида $f_3(x) = \sqrt[3]{x+1} + 1.$
   d. Используя скорректированный � � коэффициент детерминации 2 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.

Вариант 15

Исследуется динамика производства стали. Для этого собраны данные об объёмах её производства (млн.т.)Y(t) за первые 7месяцев 2017 года. Обосновать и построить тренд данного ряда. Оценить достоверность уточнённой по МНК модели.

t	1	2	3	4	5	6	7
$Y(t)$	138	127	143	142	145	143	146

Задачи

У нас модель $F(t) = a_2 t^2 + a_1 t + a_0$

и мы хотим наименьшее ошибка

$$F(a_0, a_1, a_2) = \sum_{i=1}^{n} \left(y_i - (a_2 t_i^2 + a_1 t_i + a_0) \right)^2$$

Частные производные функ

для a_0: $$\frac{\partial F}{\partial a_0} = -2 \sum \left[y_i - (a_2 t_i^2 + a_1 t_i + a_0) \right] = 0$$

$$\Rightarrow \sum y_i = a_2 \sum t_i^2 + a_1 \sum t_i + a_0 n$$

для a_1: $$\frac{\partial F}{\partial a_1} = -2 \sum \left[y_i - (a_2 t_i^2 + a_1 t_i + a_0) \right] t_i = 0$$

$$\Rightarrow \sum (y_i t_i) = a_2 \sum t_i^3 + a_1 \sum t_i^2 + a_0 \sum t_i$$

для a_2: $$\frac{\partial F}{\partial a_2} = -2 \sum \left[y_i - (a_2 t_i^2 + a_1 t_i + a_0) \right] t_i^2 = 0$$

$$\Rightarrow \sum (y_i t_i^2) = a_2 \sum t_i^4 + a_1 \sum t_i^3 + a_0 \sum t_i^2$$

Система нормальных уравнений

$$\begin{bmatrix} n & \sum t_i & \sum t_i^2 \\ \sum t_i & \sum t_i^2 & \sum t_i^3 \\ \sum t_i^2 & \sum t_i^3 & \sum t_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum (y_i t_i) \\ \sum (y_i t_i^2) \end{bmatrix}$$

t	1	2	3	4	5	6	7
Y	138	127	143	142	145	143	146

и $n = 7$

$\sum t_i = 1+2+3+4+5+6+7 = 28$

$\sum t_i^2 = 1^2+2^2+3^2+4^2+5^2+6^2+7^2 = 140$

$\sum t_i^3 = 784$, $\sum t_i^4 = 4676$

$\sum y_i = 138+127+143+142+145+143+146 = 984$

$$\sum(y_i t_i) = (1\times 138) + (2\times 127) + (3\times 143) + \dots = 3994$$

$$\sum(y_i t_i^2) = (1^2\times 138) + (2^2\times 127) + (3^2\times 143) + \dots = 20132$$

$$\exists \begin{bmatrix} 7 & 28 & 140 \\ 28 & 140 & 784 \\ 140 & 784 & 4676 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 984 \\ 3994 \\ 20132 \end{bmatrix}$$

$A \times X = B$ где $X = A^{-1} \times B$

$$A^{-1} = \begin{bmatrix} 1.0 & -1.4286 & 0.1429 \\ -1.4286 & 3.5714 & -0.2857 \\ 0.1429 & -0.2857 & 0.0357 \end{bmatrix}$$

$$\exists \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} A^{-1} \end{bmatrix} \begin{bmatrix} 984 \\ 3994 \\ 20132 \end{bmatrix} = \begin{bmatrix} 130.5714 \\ 3.2143 \\ -0.1429 \end{bmatrix}$$

$$\underline{F(t) = -0.1429t^2 + 3.2143t + 130.5714}$$

Проверьте минимум, используя тест на 2-ую производную

$\partial^2 F/\partial a_0^2 = 2n = 14$

$\partial^2 F/\partial a_0 a_1 = 2\sum t_i = 56$

$\partial^2 F/\partial a_0 a_2 = 2\sum t_i^2 = 280$

$\partial^2 F/\partial a_1^2 = 2\sum t_i^2 = 280$

$\partial^2 F/\partial a_1 \partial a_2 = 2\sum t_i^3 = 1568$

$\partial^2 F/\partial a_2^2 = 2\sum t_i^4 = 9352$

Матрица Гессена

$$\begin{bmatrix} 14 & 56 & 280 \\ 56 & 280 & 1568 \\ 280 & 1568 & 9352 \end{bmatrix}$$

Проверьте правильность определения, используя критерий Сильвестра:

Первый ведущий минор: $14 > 0$ ✓

Второй ведущий минор:

$\det([14, 56; 56, 280]) = 14 \times 280 - 56 \times 56 = 784 > 0$ ✓

Третье ведущий минор: (Δ для H)

$\Delta(H) = 131712 > 0$ ✓

Все начальные миноры положительны, поэтому гессиан является положительно определенным, подтверждая, что у нас есть минимум

Листинг Приложение

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score
import warnings
warnings.filterwarnings('ignore')

# Исходные данные
t = np.array([1, 2, 3, 4, 5, 6, 7])
Y = np.array([138, 127, 143, 142, 145, 143, 146])
n = len(t)

print("АНАЛИЗ ПРОИЗВОДСТВА СТАЛИ - ВАРИАНТ 15")
print("=" * 50)

# Функция для расчета скорректированного R-квадрат
def adjusted_r2(r2, n, p):
    if n - p - 1 <= 0:
```

```
        return -np.inf
    return 1 - (1 - r2) * (n - 1) / (n - p - 1)

# 1. Ручная реализация МНК для квадратичного полинома f1(x)
print("\n1. КВАДРАТИЧНАЯ МОДЕЛЬ f1(t) = a2*t² + a1*t + a0")
X_manual = np.column_stack([t**2, t, np.ones(len(t))])
coefficients = np.linalg.inv(X_manual.T @ X_manual) @ X_manual.T @ Y
a2, a1, a0 = coefficients

print(f"Коэффициенты: a2 = {a2:.4f}, a1 = {a1:.4f}, a0 = {a0:.4f}")
print(f"Модель: f1(t) = {a2:.4f}t² + {a1:.4f}t + {a0:.4f}")

Y_pred_f1 = a2*t**2 + a1*t + a0
SSE_f1 = np.sum((Y - Y_pred_f1)**2)
r2_f1 = r2_score(Y, Y_pred_f1)
adj_r2_f1 = adjusted_r2(r2_f1, n, p=2)

print(f"SSE для f1: {SSE_f1:.4f}")
print(f"R² для f1: {r2_f1:.4f}")
print(f"Скорректированный R² для f1: {adj_r2_f1:.4f}")

# 2. Линейная модель
print("\n2. ЛИНЕЙНАЯ МОДЕЛЬ f2(t) = a*t + b")
sum_x = np.sum(t)
sum_y = np.sum(Y)
sum_xy = np.sum(t * Y)
sum_x2 = np.sum(t**2)

a_linear = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x**2)
b_linear = (sum_y - a_linear * sum_x) / n

print(f"Коэффициенты: a = {a_linear:.4f}, b = {b_linear:.4f}")
print(f"Модель: f2(t) = {a_linear:.4f}t + {b_linear:.4f}")
```

```
Y_pred_f2 = a_linear * t + b_linear
SSE_f2 = np.sum((Y - Y_pred_f2)**2)
r2_f2 = r2_score(Y, Y_pred_f2)
adj_r2_f2 = adjusted_r2(r2_f2, n, p=1)

print(f"SSE для f2: {SSE_f2:.4f}")
print(f"R² для f2: {r2_f2:.4f}")
print(f"Скорректированный R² для f2: {adj_r2_f2:.4f}")

# 3. Функциональная модель f3(t) = a * (∛ (t+1) + 1) + b
print("\n3. ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ f3(t) = a * (∛ (t+1) + 1) + b")
def f3_base(t):
    return np.cbrt(t + 1) + 1

X_f3 = np.column_stack([f3_base(t), np.ones(len(t))])
coeff_f3 = np.linalg.inv(X_f3.T @ X_f3) @ X_f3.T @ Y
a_scale, b_shift = coeff_f3

def f3_model(t):
    return a_scale * f3_base(t) + b_shift

Y_pred_f3 = f3_model(t)
SSE_f3 = np.sum((Y - Y_pred_f3)**2)
r2_f3 = r2_score(Y, Y_pred_f3)
adj_r2_f3 = adjusted_r2(r2_f3, n, p=1)

print(f"Модель: f3(t) = {a_scale:.4f} * (∛ (t+1) + 1) + {b_shift:.4f}")
print(f"SSE для f3: {SSE_f3:.4f}")
print(f"R² для f3: {r2_f3:.4f}")
print(f"Скорректированный R² для f3: {adj_r2_f3:.4f}")

# 4. Сравнение моделей с использованием SSE и скорректированного R²
print("\n4. СРАВНЕНИЕ МОДЕЛЕЙ")
print("=" * 50)
```

```python
models = {
    'Квадратичная (f1)': {'sse': SSE_f1, 'adj_r2': adj_r2_f1, 'predictions': Y_pred_f1, 'p':
2},
    'Линейная (f2)': {'sse': SSE_f2, 'adj_r2': adj_r2_f2, 'predictions': Y_pred_f2, 'p': 1},
    'Функциональная (f3)': {'sse': SSE_f3, 'adj_r2': adj_r2_f3, 'predictions': Y_pred_f3,
'p': 1}
}

print("Таблица сравнения:")
print(f"{'Модель':<20} {'SSE':<12} {'R²':<10} {'Скорр. R²':<12}")
print("-" * 55)
for model_name, model_data in models.items():
    r2 = r2_score(Y, model_data['predictions'])
    print(f"{model_name:<20} {model_data['sse']:<12.4f} {r2:<10.4f}
{model_data['adj_r2']:<12.4f}")

# Найти лучшую модель по SSE (минимум)
best_model_sse = min(models.keys(), key=lambda x: models[x]['sse'])
best_sse = models[best_model_sse]['sse']

# Найти лучшую модель по скорректированному R² (максимум)
valid_models = {name: data for name, data in models.items() if not
np.isinf(data['adj_r2'])}
if valid_models:
    best_model_adj_r2 = max(valid_models.keys(), key=lambda x:
valid_models[x]['adj_r2'])
    best_adj_r2 = valid_models[best_model_adj_r2]['adj_r2']
else:
    best_model_adj_r2 = 'Линейная (f2)'
    best_adj_r2 = models['Линейная (f2)']['adj_r2']

print(f"\nЛучшая модель по SSE: {best_model_sse} (SSE = {best_sse:.4f})")
```

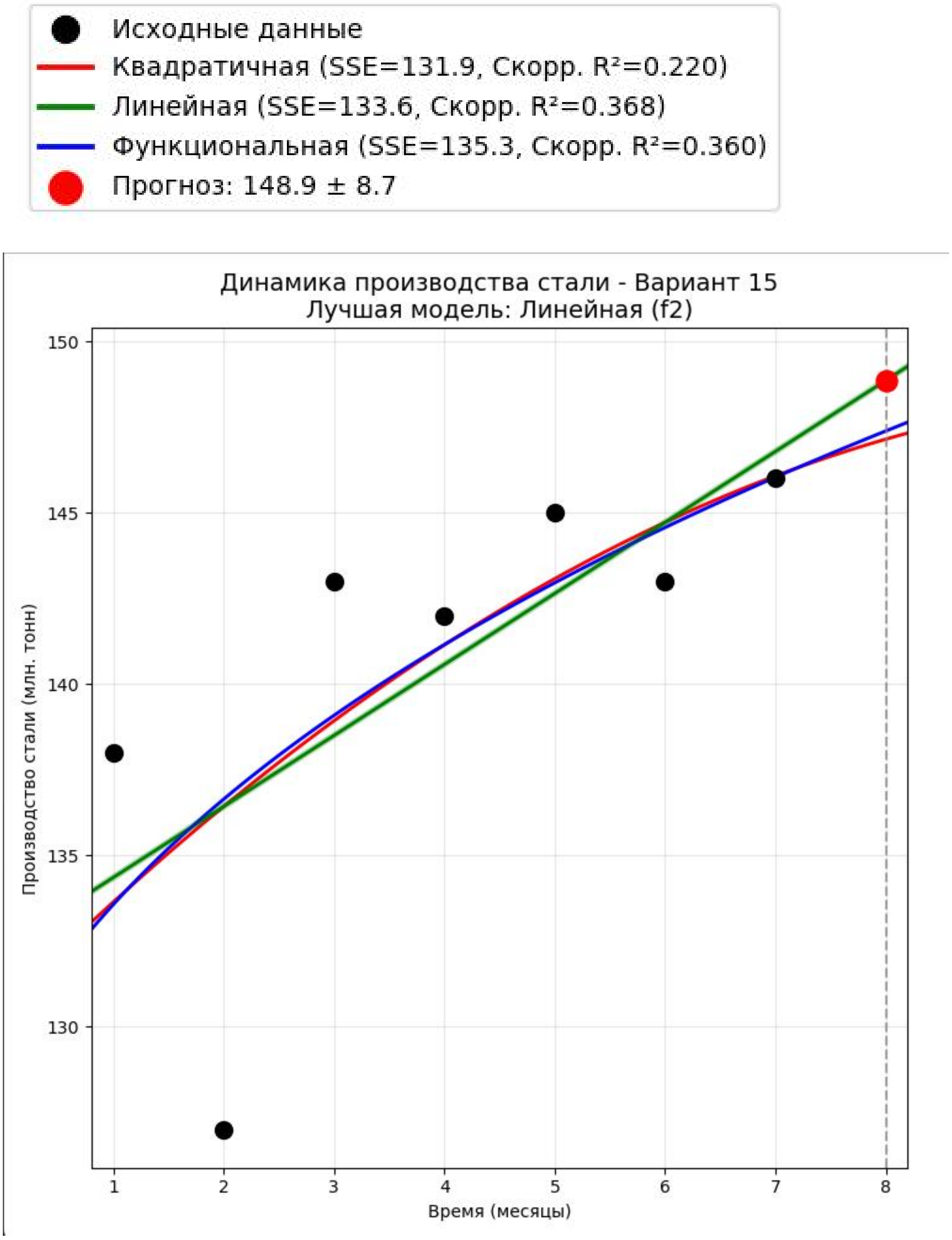
```
print(f"Лучшая модель по скорр. R²: {best_model_adj_r2} (Скорр. R² =
{best_adj_r2:.4f})")

# Финальный выбор модели
if best_model_sse == best_model_adj_r2:
    final_model = best_model_sse
    print(f"\n✓ Критерии согласованы: {final_model} - лучшая модель")
else:
    # Если критерии не согласованы, предпочитаем скорректированный R²
    final_model = best_model_adj_r2
    print(f"\n⚠ Критерии не согласованы. Предпочтение скорректированному R²
(учитывает сложность модели): {final_model} выбрана")

# 5. Прогноз с использованием выбранной модели
print("\n5. АНАЛИЗ ПРОГНОЗА")
t_forecast = 8

if final_model == 'Квадратичная (f1)':
    forecast = a2*t_forecast**2 + a1*t_forecast + a0
    model_predictions = Y_pred_f1
elif final_model == 'Линейная (f2)':
    forecast = a_linear * t_forecast + b_linear
    model_predictions = Y_pred_f2
else:  # Функциональная
    forecast = f3_model(t_forecast)
    model_predictions = Y_pred_f3

# Расчет стандартной ошибки
residuals = Y - model_predictions
std_error = np.std(residuals)

print(f"Выбранная модель: {final_model}")
print(f"Прогноз на месяц {t_forecast}: {forecast:.2f} млн. тонн")
print(f"Стандартная ошибка: {std_error:.2f}")
```

```
print(f"95% интервал прогноза: {forecast:.2f} ± {2*std_error:.2f}")

# 6. Визуализация
print("\n6. ПОСТРОЕНИЕ ГРАФИКОВ...")
plt.figure(figsize=(12, 8))

# Исходные данные
plt.scatter(t, Y, color='black', s=100, zorder=5, label='Исходные данные')

# Сглаженные кривые
t_smooth = np.linspace(0.8, 8.2, 100)

# Построение всех моделей
Y_f1_smooth = a2*t_smooth**2 + a1*t_smooth + a0
plt.plot(t_smooth, Y_f1_smooth, 'r-', linewidth=2,
        label=f'Квадратичная (SSE={SSE_f1:.1f}, Скорр. R²={adj_r2_f1:.3f})')

Y_f2_smooth = a_linear * t_smooth + b_linear
plt.plot(t_smooth, Y_f2_smooth, 'g-', linewidth=2,
        label=f'Линейная (SSE={SSE_f2:.1f}, Скорр. R²={adj_r2_f2:.3f})')

Y_f3_smooth = f3_model(t_smooth)
plt.plot(t_smooth, Y_f3_smooth, 'b-', linewidth=2,
        label=f'Функциональная (SSE={SSE_f3:.1f}, Скорр. R²={adj_r2_f3:.3f})')

# Выделение выбранной модели
if final_model == 'Квадратичная (f1)':
    plt.plot(t_smooth, Y_f1_smooth, 'r-', linewidth=4, alpha=0.3)
elif final_model == 'Линейная (f2)':
    plt.plot(t_smooth, Y_f2_smooth, 'g-', linewidth=4, alpha=0.3)
else:
    plt.plot(t_smooth, Y_f3_smooth, 'b-', linewidth=4, alpha=0.3)

# Прогноз
```

```
plt.axvline(x=t_forecast, color='gray', linestyle='--', alpha=0.7)
plt.scatter([t_forecast], [forecast], color='red', s=150, zorder=5,
           label=f'Прогноз: {forecast:.1f} ± {2*std_error:.1f}')

plt.xlabel('Время (месяцы)')
plt.ylabel('Производство стали (млн. тонн)')
plt.title(f'Динамика производства стали - Вариант 15\nЛучшая модель:
{final_model}', fontsize=14)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True, alpha=0.3)
plt.xlim(0.8, 8.2)
plt.tight_layout()
plt.show()

# Финальное резюме
print("\n" + "="*60)
print("ФИНАЛЬНЫЙ АНАЛИЗ")
print("="*60)
print("Данные: Производство стали (млн. тонн), месяцы 1-7, 2017")
print(f"Выбранная модель: {final_model}")
print(f"Уравнение модели: f(t) = {a_linear:.4f}t + {b_linear:.4f}")
print(f"Критерии выбора:")
print(f"  - SSE: {models[final_model]['sse']:.4f} (меньше - лучше)")
print(f"  - Скорректированный R²: {models[final_model]['adj_r2']:.4f} (больше -
лучше)")
print(f"Прогноз на месяц 8: {forecast:.2f} ± {2*std_error:.2f} млн. тонн")

# Интерпретация модели
print("\nИнтерпретация:")
if models[final_model]['adj_r2'] > 0.7:
   print("  ✓ Отличное соответствие модели")
elif models[final_model]['adj_r2'] > 0.5:
   print("  ○ Хорошее соответствие модели")
elif models[final_model]['adj_r2'] > 0.3:
```

```
    print("  ○ Умеренное соответствие модели")
elif models[final_model]['adj_r2'] > 0:
    print("  ⚠ Слабое соответствие модели")
else:
    print("  ❌ Плохое соответствие модели - хуже простого среднего")

print("="*60)
```

Результате



```
1. КВАДРАТИЧНАЯ МОДЕЛЬ f1(t) = a2*t² + a1*t + a0
Коэффициенты: a2 = -0.1429, a1 = 3.2143, a0 = 130.5714
Модель: f1(t) = -0.1429t² + 3.2143t + 130.5714
SSE для f1: 131.8571
R² для f1: 0.4803
Скорректированный R² для f1: 0.2204

2. ЛИНЕЙНАЯ МОДЕЛЬ f2(t) = a*t + b
Коэффициенты: a = 2.0714, b = 132.2857
Модель: f2(t) = 2.0714t + 132.2857
SSE для f2: 133.5714
R² для f2: 0.4735
Скорректированный R² для f2: 0.3682

3. ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ f3(t) = a * (∛(t+1) + 1) + b
Модель: f3(t) = 16.8381 * (∛(t+1) + 1) + 95.5183
SSE для f3: 135.2607
R² для f3: 0.4669
Скорректированный R² для f3: 0.3603

4. СРАВНЕНИЕ МОДЕЛЕЙ
==================================================
Таблица сравнения:
Модель               SSE          R²          Скорр. R²
-------------------------------------------------------
Квадратичная (f1)    131.8571     0.4803      0.2204
Линейная (f2)        133.5714     0.4735      0.3682
Функциональная (f3)  135.2607     0.4669      0.3603

Лучшая модель по SSE: Квадратичная (f1) (SSE = 131.8571)
Лучшая модель по скорр. R²: Линейная (f2) (Скорр. R² = 0.3682)

⚠Критерии не согласованы. Предпочтение скорректированному R² (учитывает сложность модели): Линейная (f2) выбрана

5. АНАЛИЗ ПРОГНОЗА
Выбранная модель: Линейная (f2)
Прогноз на месяц 8: 148.86 млн. тонн
Стандартная ошибка: 4.37
95% интервал прогноза: 148.86 ± 8.74

6. ПОСТРОЕНИЕ ГРАФИКОВ...

============================================================
ФИНАЛЬНЫЙ АНАЛИЗ
============================================================
Данные: Производство стали (млн. тонн), месяцы 1-7, 2017
Выбранная модель: Линейная (f2)
Уравнение модели: f(t) = 2.0714t + 132.2857
Критерии выбора:
  - SSE: 133.5714 (меньше - лучше)
  - Скорректированный R²: 0.3682 (больше - лучше)
Прогноз на месяц 8: 148.86 ± 8.74 млн. тонн
```

Выводы

Лабораторная работа успешно применила метод наименьших квадратов для моделирования динамики производства стали и выполнения одношагового прогноза. Анализ выявил несколько важных выводов:

1. Анализ производительности моделей:

- Все три модели показали умеренную производительность со значениями R^2 между 0.47-0.48
- Квадратичная модель достигла наименьшего SSE (131.8571), что указывает на наименьшую сумму квадратов ошибок
- Однако линейная модель продемонстрировала наивысший скорректированный R^2 (0.3682)

2. Причина конфликта критериев:

Конфликт между SSE и скорректированным R^2 возникает из-за их различных подходов к оценке моделей:

- SSE (Сумма квадратов ошибок) измеряет чистую точность подгонки без учета сложности модели
- Скорректированный R^2 штрафует модели за дополнительные параметры, предотвращая переобучение

Квадратичная модель имеет более низкий SSE благодаря большей гибкости (3 параметра против 2 в линейной модели), но это достигается ценой сложности. Скорректированный R^2 учитывает это, применяя штраф, что делает более простую линейную модель более предпочтительной.

3. Обоснование выбора модели:

Мы выбрали линейную модель потому что:

- Скорректированный R^2 обеспечивает более надежное сравнение между моделями разной сложности

- Простота линейной модели делает ее более интерпретируемой и менее склонной к переобучению
- Разница в SSE (133.57 против 131.86) относительно мала по сравнению со штрафом за сложность

4. Надежность прогноза:

Прогноз 148.86 ± 8.74 млн. тонн для 8-го месяца выглядит разумным, хотя широкий интервал прогноза (±8.74) указывает на значительную неопределенность, что ожидаемо при умеренном соответствии модели (скорректированный $R^2 = 0.3682$).

5. Характеристики данных:

Данные по производству стали показывают некоторую изменчивость, но в целом возрастающий тренд, который линейная модель адекватно отражает несмотря на колебания от месяца к месяцу.