|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«*Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ: 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА\_\_\_

**ОТЧЕТ**

**по лабораторной работе №** \_\_1**\_\_**

**Дисциплина: Прикладной анализ данных**

**Название лабораторной работы:** Прогнозирование моделью линейной регрессии

Студент гр. ИУ6-55Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  А.М. Латышев

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2024

**Цель работы:**

Изучить методы построения линейных регрессионных моделей с использованием аналитического и итерационного метода. Произвести прогнозирование моделью линейной регрессии.

**Задание:**

Вариант 12

Рассмотрим пример построения линейной регрессионной модели на основе информации об ожидаемой продолжительности жизни мужчин, число лет (у), рождаемости населения на 1000 человек (х1), смертности населения на 1000 человек (х2), числе браков на 1000 человек (х3), числе разводов на 1000 человек (х4), коэффициенте младенческой смертности (х5), соотношении денежного дохода и прожиточного минимума, % (х6), соотношении средней оплаты труда и прожиточного минимума трудоспособного населения, % (х7), численности населения с денежными доходами ниже прожиточного минимума в % от численности населения (х8), числа зарегистрированных преступлений на 100000 населения (х9).

Этапы:

1. Нормирование (масштабирование) исходных данных.
2. Расчет весов линейной регрессии по аналитической формуле – предлагается использовать numpy.
3. Построение и интепретация корреляционной матрицы. Определение степени мультиколлинеарности на основе числа обусловленности. Для их расчета не нужно писать самописные методы, предлагается использовать scipy.
4. Анализ регрессионных остатков.
5. Определение весов линейной регрессии градиентным методом – предлагается использовать numpy. Проанализировать изменение ошибки от итерации к итерации.
6. Сравнение результатов по аналитическому и градиентному методу.
7. С помощью библиотеки sklearn сделать fit-predict модели линейной регрессии. Сравнить результаты с ранее полученными.
8. С помощью библиотеки statmodels получить «эконометрический» результат обучения модели линейной регрессии. Проинтерпретировать все его составляющие (в т.ч. те, которые изучались только теоретически), сравнить с предыдущими результатами. Пример отчета - https://www.statsmodels.org/stable/index.html P.S. Регрессионным статистическим гипотезам будет уделено особое внимание на защите.
9. Сравнить качество получаемых моделей на основе коэффициента детерминации и MSE.
10. Сделать итоговый вывод касательно причин различия в результатах при выполнении работ разными методами, а также по получаемым моделям в целом. Провести сравнительный анализ.

**Ход работы:**

1. Исходные данные были нормированы (масштабированы) для улучшения процесса обучения модели.

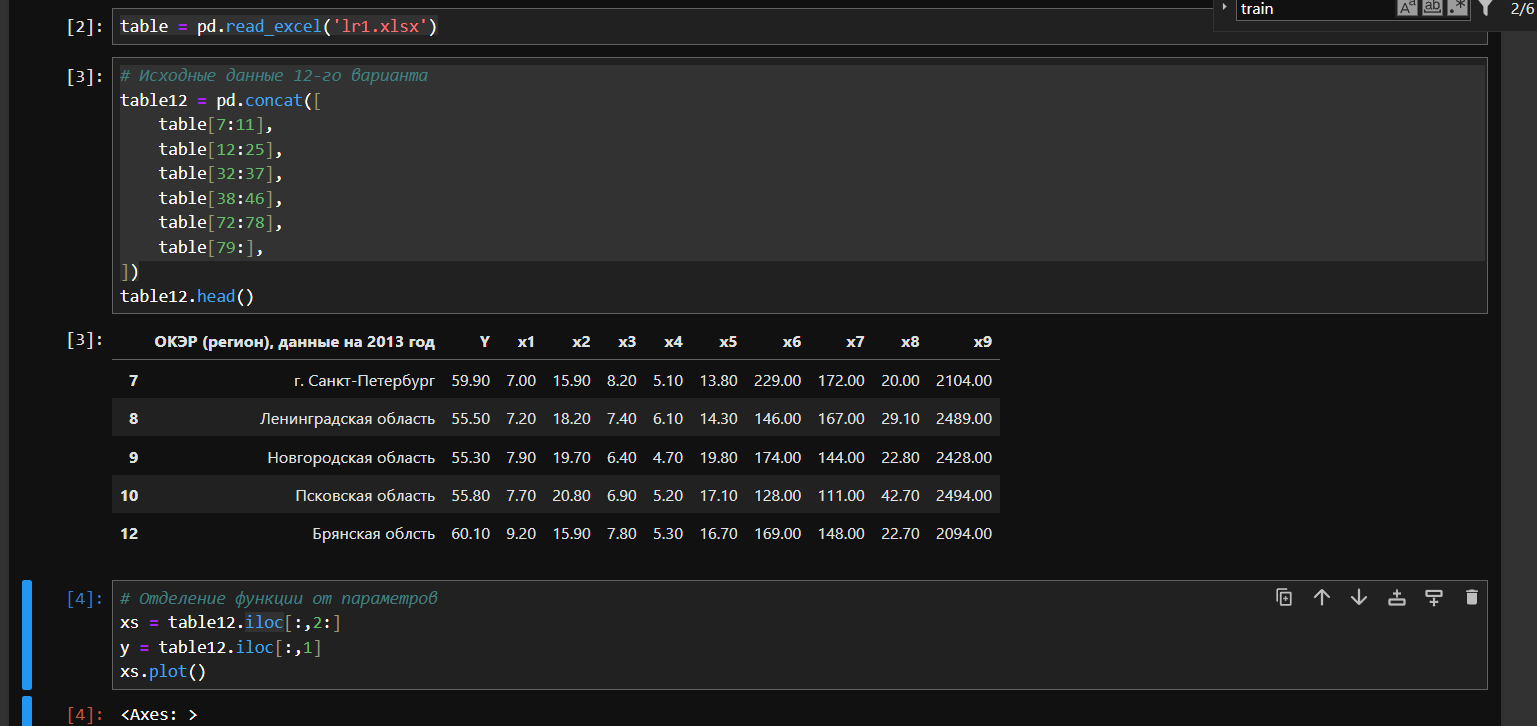
Была выполнена выгрузка данных из исходной таблицы xlsx, предложенной преподавателем и выбраны только необходимые строки, соответствующие 12-му варианту (см. Рисунок 1).

Данные всех параметров x были отображены на одном графике (см. Рисунок 2). На примере x9 хорошо заметно, что данные по каждому конкретному параметру распределены не нормально, так же заметно, что параметры не нормированы относительно друг друга.

В данном случае отлично справится с задачей нормалищации Z-преобразование (z-score normalization).

Используя стандартное отклонение(x.std()) и среднее значение(x.mean()) на наборе данных, масштабы и единицы измерения были приведены к единому виду (Рисунок 3).

Таким образом упрощен предстоящий анализ данных.

  
Рисунок 1 – Исходные данные по варианту

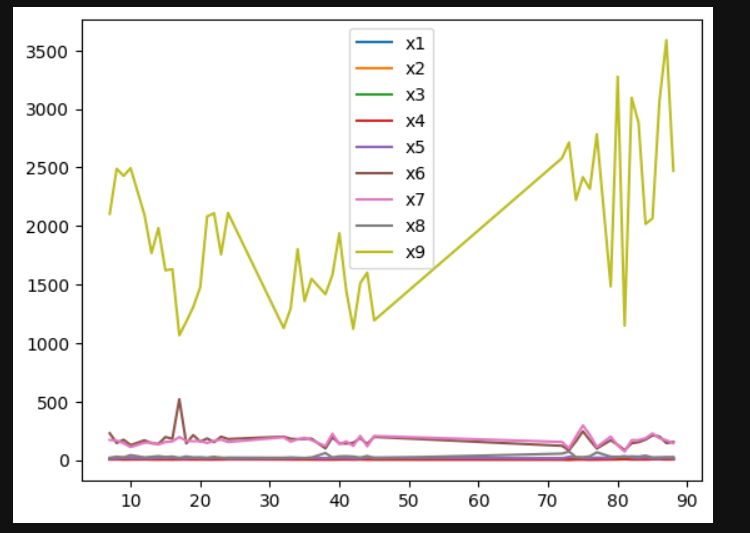


Рисунок 2 – График на основе исходных данных

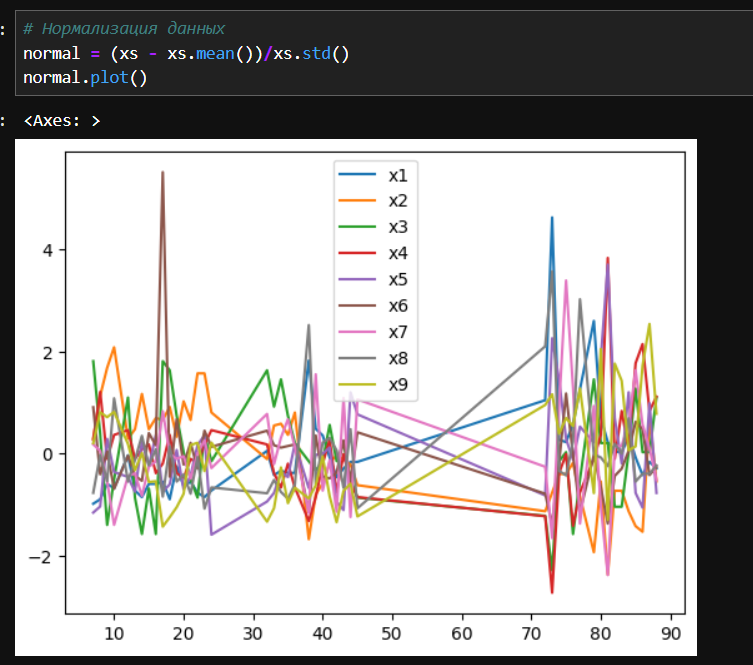


Рисунок 3 – Нормализация данных

1. Были рассчитаны веса линейной регрессии по аналитической формуле с использованием библиотеки numpy.

Расчет весов по аналитической формуле подразумевает вычисление ковариационной матрицы на данных после добавления единичного столбца (normal.T.dot(normal)) и вычисление псевдообратной матрицы на ее основе.

Процесс вычисления весов и результат приведены на Рисунке 4.



Рисунок 4 – Веса линейной регрессии

1. Была построена корреляционная матрица, определена степень мультиколлинеарности на основе числа обусловленности с помощью библиотеки scipy.

С использованием библиотеки scipy нашли корреляционную матрицу и число обусловленности (см. Рисунок 5).

С помощью градиента получаем тепловую карту.

Исходные данные обладают статистической зависимостью, поэтому наблюдаем частичную мультиколлинеарность (значения модуля корреляции 0.6 и более между разными признаками).

Проанализируем высокую корреляцию между признаками:

* Рождаемость населения на 1000 человек (х1) коррелирует с численностью населения с денежными доходами ниже прожиточного минимума в % от численности населения (х8).
* Рождаемость населения на 1000 человек (х1) обратно коррелирует со смертностбю населения на 1000 человек (х2).



Рисунок 5 – Корреляционная матрица

1. Был проведен анализ регрессионных остатков, чтобы оценить качество модели.

Вычисление регрессионных остатков(ошибки) приведено на Рисунке 6. Также была вычислена средняя квадратическая ошибка (1.13) и коэффициент детерминации.

Для оценки качества модели необходимо использовать теорему Гаусса-Маркова.

* Все x детерминированны
* Признаки линейно не зависимы
* Мат ожидание ошибки равно нулю (5 \* 10^-15)
* Дисперсия равна константе (1.159)
* Отсутствует автокорреляция

На рисунке 7 приведен график распределения ошибки и нормальное распределение.

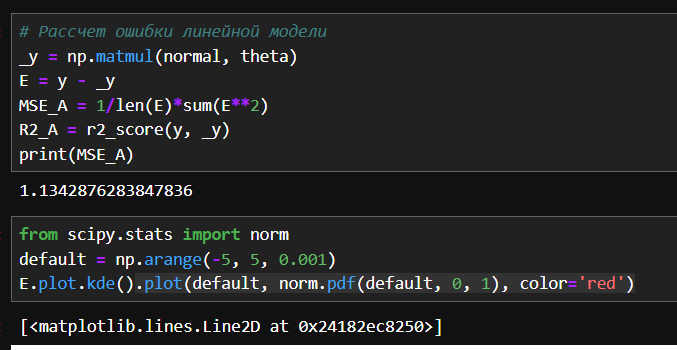


Рисунок 6 – анализ регрессионных остатков

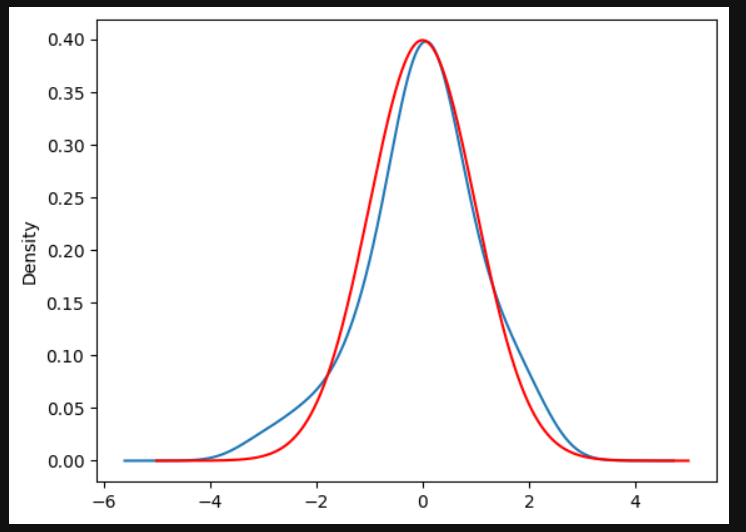


Рисунок 7 – График распределения ошибки

1. Были определены веса линейной регрессии градиентным методом с использованием библиотеки numpy, и было проанализировано изменение ошибки от итерации к итерации.

Для применения метода градиентного спуска необходимо было:

1. Инициализировать начальные веса (1), скорость обучения (0.01) и количество итераций (10000).
2. Итеративно рассчитать предсказанный y и градиент функции потерь, считать новые веса.

Алгоритм расчета приведен на рисунке 8.

Самым показательным является изменение ошибки на первой тысяче итераций. Далее производная стремится к нулю, т.к. приближаемся к целевому значению ошибки.

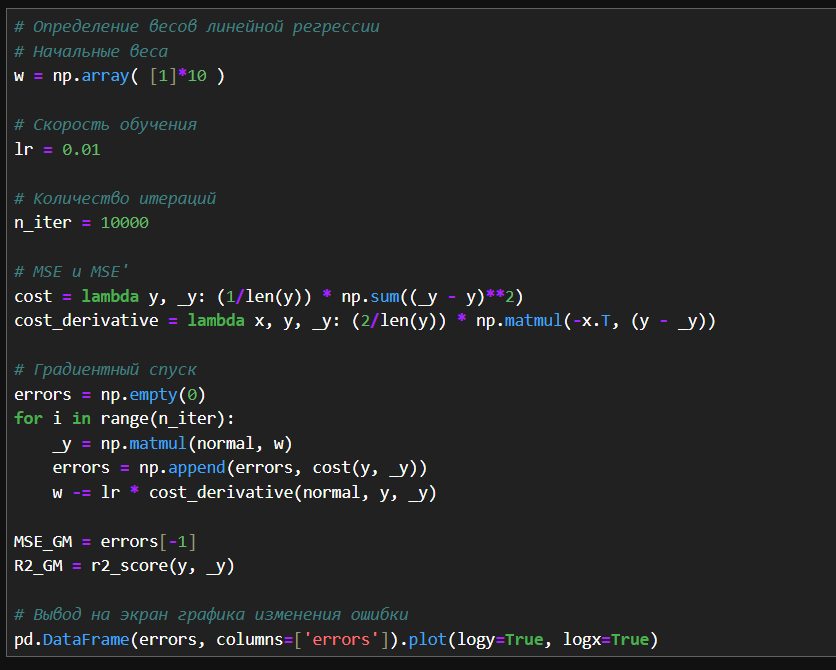


Рисунок 8 – Алгоритм расчета весов итерационным методом

1. Было произведено сравнение результатов по аналитическому и градиентному методу.

Результаты полученные аналитическим методов оказались очень точными, т.к. расхождение появляется только в 15 знаке после запятой, учитывая, что в итеративном методе было произведено большое количество итераций, что затратило большое количество вычислительных ресурсов.

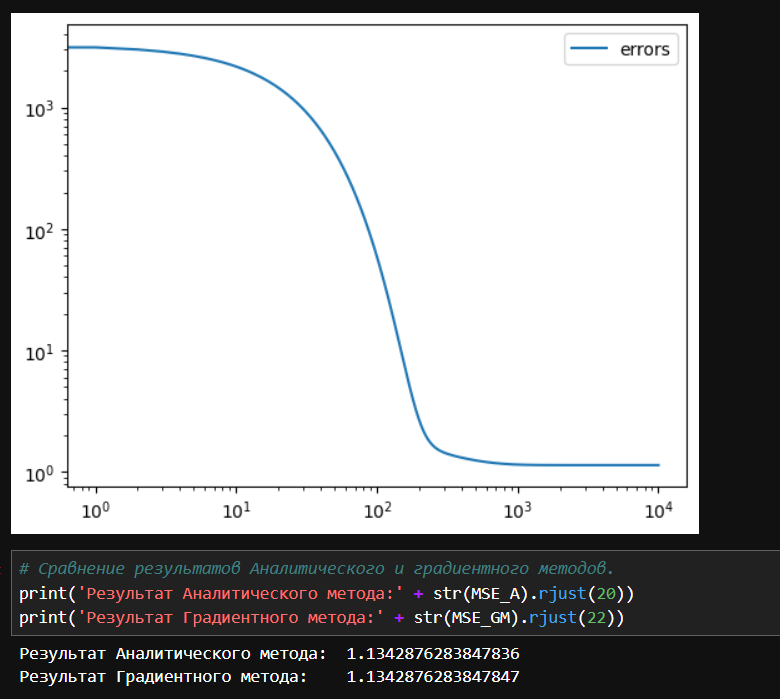


Рисунок 9 – Результат обучения модели и результаты.

1. Была использована библиотека sklearn для создания модели линейной регрессии с помощью метода fit-predict, и было произведено сравнение результатов с предыдущими.

Для корректной проверки модели линейной регрессии разделили данные на тренировочные и тестовые в соотношении 80/20. Обучение данных происходило только на тренировочных данных, а проверка, только на тестовых.

Среднеквадратичная ошибка полностью совпала с результатами градиентного спуска.

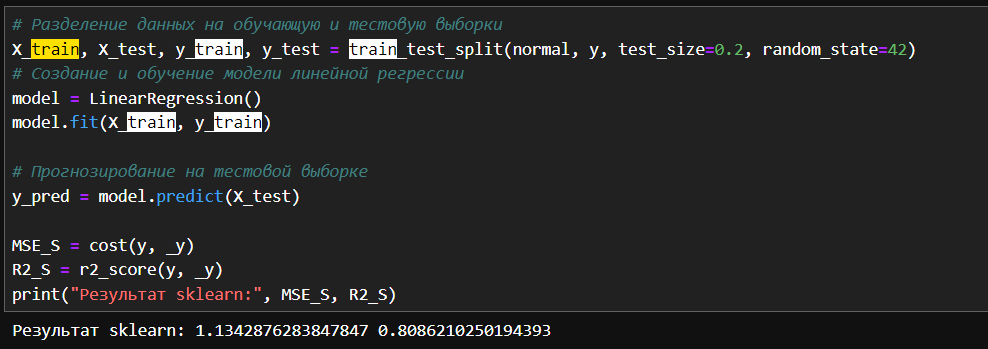


Рисунок 10 – Результат использования модели линейной регрессии

1. С помощью библиотеки statmodels был получен "эконометрический" результат обучения модели линейной регрессии, и были проинтерпретированы все его составляющие.

С использованием библиотеки получили полную таблицу параметров модели.

Зависимая переменная – Y

Модель Метод наименьших квадратов – OLS

Метод – наименьших квадратов

Дата, Время

Количество наблюдений (строк исходной таблицы) – 46

Degrees of Freedom – количество наблюдений минус количество прогнозов минус 1 - 36

Количество прогнозируемых переменных – 9 (количество Y)

Тип ковариации – неустойчива к гомоскедастичности

Коэффициент детерминации – 0.809

Скорректированный коэфициент детерминации – 0.761

F-тест или критерий Фишера – 16.9

Вероятность того, что нулевая гипотеза окажется верной – около 0

Логарифм правдоподобия - -68.169

Информационный критерий Акаике (AIC) представляет собой оценку ошибки прогнозирования – 156.3

Информационный критерий Шварца является критерием выбора модели среди конечного набора моделей – 174.6

В таблице признаков перечислены веса, стандартное отклонение ошибки, коэффициент стьюдента, вероятность, что равен нулю и интервалы коэффициента стьюдента.

Коэффициент нормальности распределения остатков - 3.174

Cтатистический тест, измеряющий вероятность нормального распределения остатков – 0.205

Мера симметрии распределения остатков - -0.453

Концентрация распределения остатков – 3.524

Критерий наличия автокорреляции – 1.825

JB Альтернативный метод Omnibus – 2.102

Prob( JB ) Альтернативный метод Prob( Omnibus ) – 0.350

Число обусловленности – 5.56

Вычисляемые прежде значения другими моделями совпали с высокой точностью.

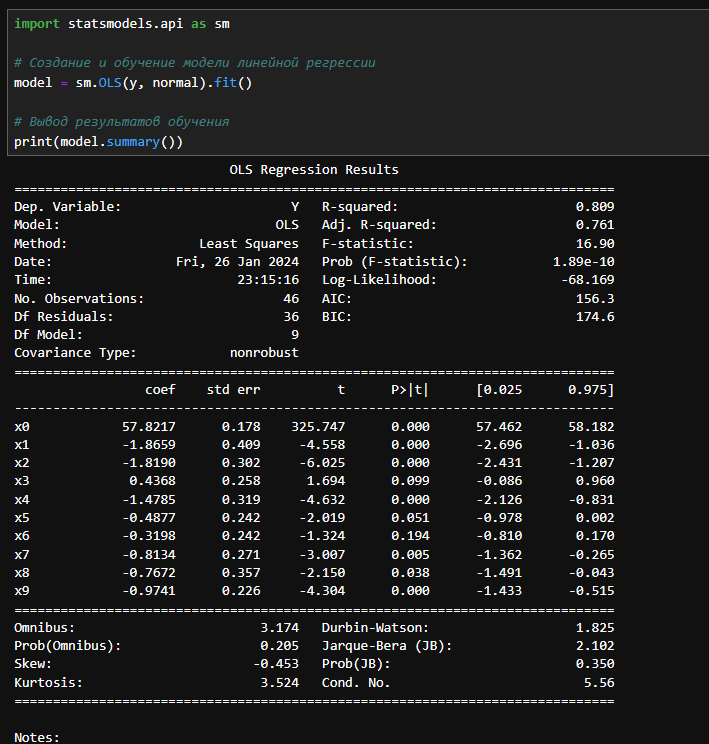


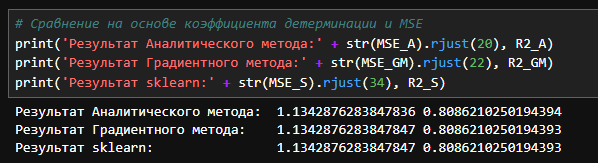
Рисунок 11 – Использование библиотеки statmodels

1. Было проведено сравнение качества получаемых моделей на основе коэффициента детерминации и MSE.

Утверждение о высокой точности Аналитического метода в пункте 6 оправдалось, т.к. без высоких затрат вычислительных ресурсов был получен результат совпадающий вплоть до 14 знака после запятой.

Однако использование библиотеки sklearn оказалось более точным, т.к. полностью совпало с результатом градиентного метода.

При этом же скорость вычислений оказалась намного выше градиентного спуска, вероятно модель выставляет оптимальные параметры при обучении для достижения точности в 16 знаков после запятой. Точность модели градиентного спуска на 10к итераций могла оказаться еще выше.

  
Рисунок 11 – Сравнение моделей

1. В итоге был сделан вывод о том, что различия в результатах при выполнении работ разными методами могут быть связаны с разными параметрами модели, нормализацией данных, выбором метода оптимизации, а также с тем, как были обработаны данные. Был проведен сравнительный анализ полученных моделей и выбрана наилучшая модель для дальнейшего использования.

Таким образом оптимальной моделью оказалась модель линейной регрессии на fit-predict методе с использованием библиотеки sklearn. Она обладает высокой скоростью вычисления, при этом показывает высокую точность.

**Вывод:**

В ходе выполнения лабораторной работы были изучены методы построения линейных регрессионных моделей с использованием аналитического и итерационного метода. Был проведен сравнительный анализ моделей, в ходе которого была выбрана наилучшая модель для дальнейшего использования.