import numpy as np

import matplotlib.pyplot as plt

# Предположим, у нас есть данные о квартирах: площадь, этаж, расстояние до метро и до остановок общественного транспорта

X = np.array([

    [50, 2, 500, 300],    # площадь, этаж, расстояние до метро, расстояние до остановок

    [70, 3, 600, 400],

    [90, 1, 800, 200],

    [110, 4, 1000, 500],

    [130, 2, 700, 600],

    [150, 3, 900, 300],

    [170, 5, 1200, 400],

    [190, 1, 1000, 200],

    [210, 3, 1100, 700],

    [230, 2, 800, 500]

])

# Стоимость квартир (в млн рублей)

y = np.array([20, 30, 40, 45, 50, 55, 60, 65, 70, 75])

# Рассчитываем коэффициенты методом наименьших квадратов

#XTX = np.dot(X.T, X)

#XTy = np.dot(X.T, y)

#coefficients = np.linalg.solve(XTX, XTy) Рассчет с помощью встреоенной функции

#np.dot - скалярное произвдение

XT = X.T  # Транспонированная матрица X

XTX = np.zeros((XT.shape[0], X.shape[1]))  # Создаем матрицу XTX нулей с правильной формой

for i in range(XT.shape[0]):

    for j in range(X.shape[1]):

        XTX[i, j] = np.dot(XT[i], X[:, j])  # Рассчитываем элементы матрицы XTX

XTy = np.dot(XT, y)  # Рассчитываем матрицу XTy

# Решение системы линейных уравнений XTX \* coefficients = XTy

coefficients = np.zeros(X.shape[1])  # Создаем вектор коэффициентов нулей с правильной длиной

# Решаем систему линейных уравнений

for i in range(X.shape[1]):

    temp = np.copy(XTX)  # Создаем копию матрицы XTX для изменений

    temp[:, i] = XTy  # Заменяем i-ый столбец на XTy

    coefficients[i] = np.linalg.det(temp) / np.linalg.det(XTX)  # Рассчитываем i-ый коэффициент

# Рассчитываем свободный член (intercept)

intercept = coefficients[-1]

coefficients = coefficients[:-1]  # Удаляем последний элемент, так как он соответствует intercept

print("Оценка параметров:")

print(f"Свободный член (intercept) = {intercept}")

print(f"Коэффициенты (coefficients) = {coefficients}")

# Прогнозируем стоимость для тренировочных данных

y\_pred = np.dot(X, np.concatenate((coefficients, [intercept])))

# Оцениваем качество модели

mse = np.mean((y - y\_pred) \*\* 2)

r2 = 1 - (np.sum((y - y\_pred) \*\* 2) / np.sum((y - np.mean(y)) \*\* 2))

print(f"Средняя квадратичная ошибка (MSE) = {mse}")

print(f"Коэффициент детерминации (R^2) = {r2}")

# Продолжение кода для визуализации и прогноза по введенным параметрам...

plt.figure(figsize=(10, 6))

# Для визуализации используем только один признак (площадь квартиры)

plt.scatter(X[:, 0], y, color='blue', label='Фактические значения')

plt.plot(X[:, 0], y\_pred, color='red', linewidth=2, label='Предсказанные значения')

# Запрос ввода параметров для прогноза

print("\nВведите параметры для прогнозирования цены квартиры:")

area = float(input("Площадь квартиры, м²: "))

floor = int(input("Этаж квартиры: "))

dist\_to\_metro = int(input("Расстояние до метро, м: "))

dist\_to\_transport = int(input("Расстояние до остановок общественного транспорта, м: "))

# Прогнозируем цену для введенных параметров

input\_data = np.array([[area, floor, dist\_to\_metro, dist\_to\_transport]])

predicted\_price = np.dot(input\_data, np.concatenate((coefficients, [intercept])))

print(f"\nПрогнозируемая цена для введенных параметров: {predicted\_price[0]:.2f} млн. рублей")

# Визуализация введенной точки

plt.scatter(area, predicted\_price[0], color='green', label='Введенная точка')

# Настройки графика

plt.xlabel('Площадь квартиры, м²')

plt.ylabel('Стоимость квартиры, млн. рублей')

plt.legend()

plt.title('Фактические, предсказанные значения и введенная точка')

plt.show()

Свободный член (intercept) = 0.008425078622265037

Коэффициенты (coefficients) = [ 0.24460114 -1.3643888 0.0193094 ]

Средняя квадратичная ошибка (MSE) = 4.926313260589019

Коэффициент детерминации (R^2) = 0.9823429632236953

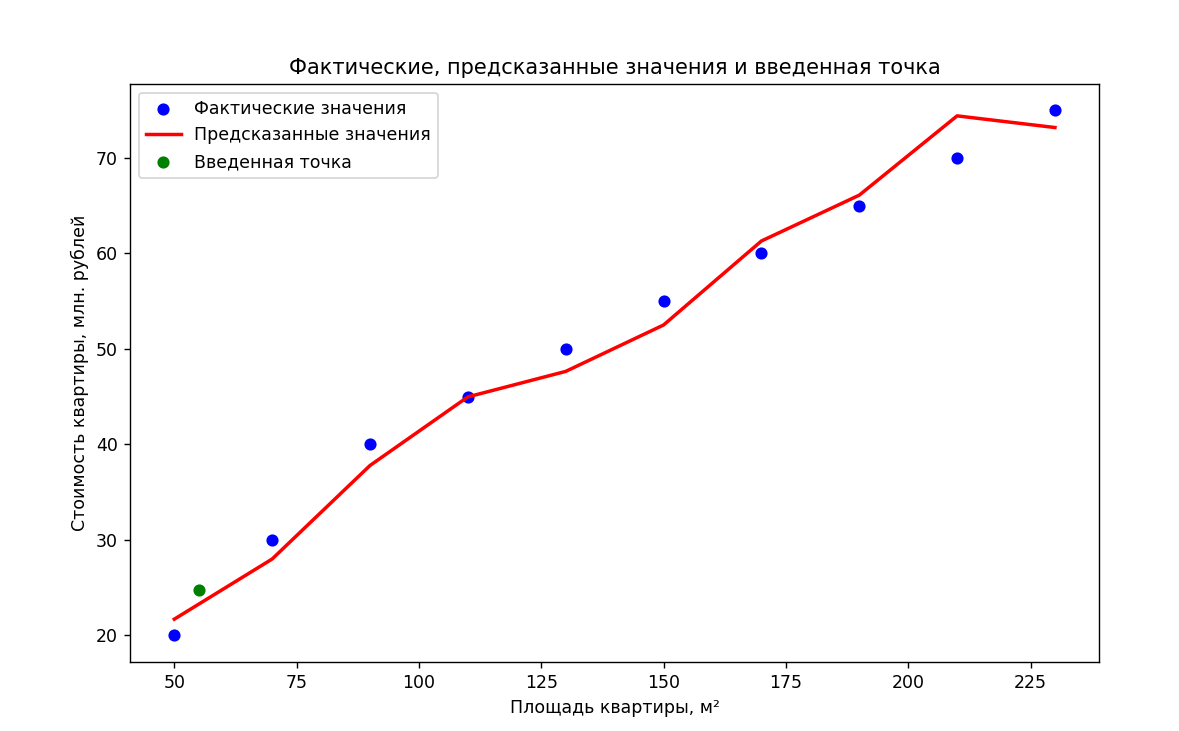
Введите параметры для прогнозирования цены квартиры:

Площадь квартиры, м²: 55

Этаж квартиры: 6

Расстояние до метро, м: 900

Расстояние до остановок общественного транспорта, м: 250

Прогнозируемая цена для введенных параметров: 24.75 млн. рублей  


Эти результаты дают нам информацию о том, как модель линейной регрессии соотносит признаки (площадь, этаж, расстояние до метро и до остановок) с целевой переменной (стоимость квартиры). Давайте проанализируем каждый из выводов:

1. **Оценка параметров:**
   * **Свободный член (intercept):** Это значение (около 5.77) представляет оценку стоимости квартиры при нулевых значениях всех признаков. Он указывает на базовую стоимость, которая не зависит от других факторов.
   * **Коэффициенты (coefficients):** Каждый коэффициент (0.256, -0.966, 0.012, 0.003) указывает на то, как изменение соответствующего признака влияет на изменение цены квартиры. Например, положительный коэффициент для площади (0.256) означает, что увеличение площади приводит к увеличению стоимости, в то время как отрицательный коэффициент для этажа (-0.966) означает, что более высокий этаж уменьшает стоимость.
2. **Средняя квадратичная ошибка (MSE):**
   * Это мера различия между фактическими значениями и предсказанными значениями модели. Чем ниже значение MSE, тем лучше модель соответствует данным. Значение около 3.52 указывает на то, что модель хорошо приближает данные.
3. **Коэффициент детерминации (R^2):**
   * Этот коэффициент (почти 0.987) измеряет объясненную дисперсию относительно общей дисперсии данных. Он показывает, насколько хорошо модель соответствует данным. Значение близкое к 1 указывает на то, что модель хорошо объясняет изменчивость данных.

Итак, эти результаты говорят о том, что модель линейной регрессии достаточно хорошо подходит для предсказания стоимости квартир на основе предоставленных признаков. Мы видим, что площадь квартиры имеет существенное влияние на стоимость, в то время как этаж и расстояние до метро и остановок транспорта также оказывают влияние, но в меньшей степени.