|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра прикладной математики (ПМ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6**

по дисциплине «Языки программирования для статистической обработки данных»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИМБО-11-23, Журавлев Ф.А.* | (подпись) | |  |
| Преподаватель | *Трушин СМ* | (подпись) | |  |
|  |  | |  | |

Москва 2025 г.

## 1) ЦЕЛЬ И ЗАДАЧИ

**Цель практической работы:**

Освоить построение моделей множественной регрессии, интерпретацию их результатов и визуализацию метрик качества в Python, R.

**Задачи практической работы:**

1. Построить модели множественной регрессии:

• Выявить влияние нескольких факторов на целевую переменную.

• Python: использование statsmodels и sklearn для построения моде лей.

• R: использование функции lm().

2. Оценить качество моделей:

• Рассчитать метрики R² и MSE для оценки качества модели.

• Python: инструменты sklearn.metrics.

• R: функции summary() и пакеты для анализа ошибок.

3. Визуализировать результаты:

• Построение графиков зависимости предсказанных значений от реальных.

• Сравнить удобство визуализации в Python и R.

4. Провести сравнительный анализ результатов, полученных в трёх инструментах.

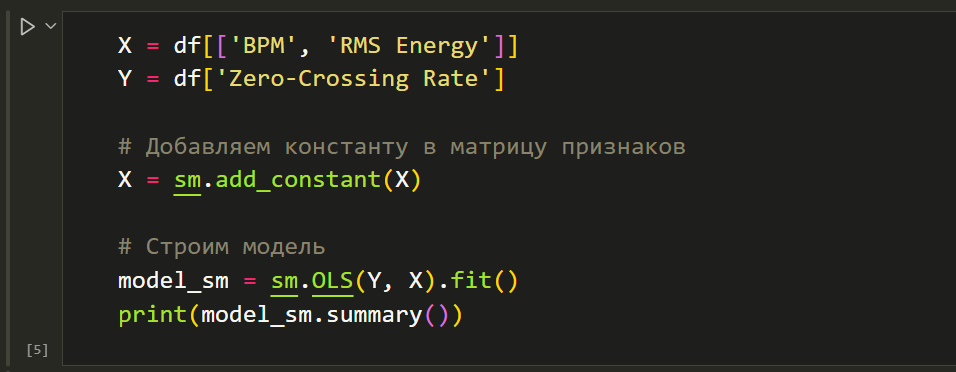
## 2) РЕЗУЛЬТАТЫ ПРАКТИКИ

**Шаг 1) Множественная линейная регрессия в Python**

### 1.1) Построение множественной линейной регрессии.

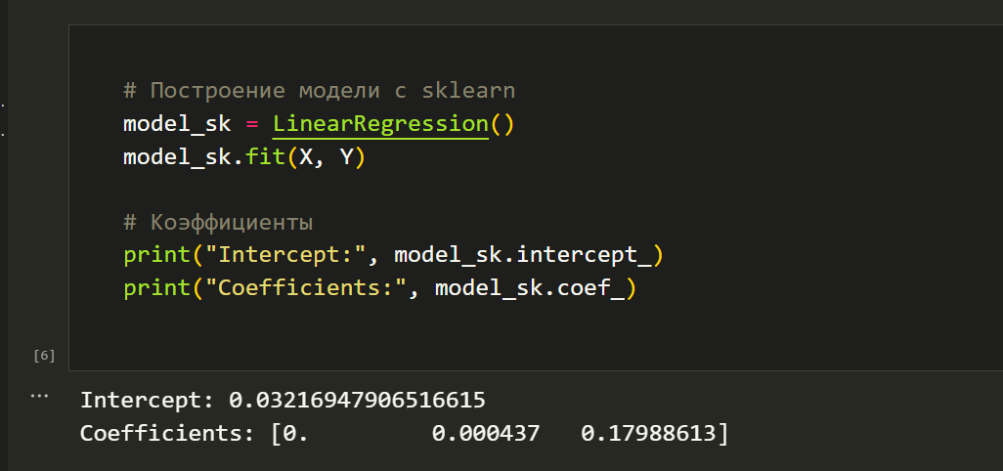
После загрузки исходной таблицы данных в формате .csv, следует написать код для построения множественной линейной регрессии. Воспользуемся двумя способами.

*Рисунок 1.1 — Построение множественной линейной регрессии*

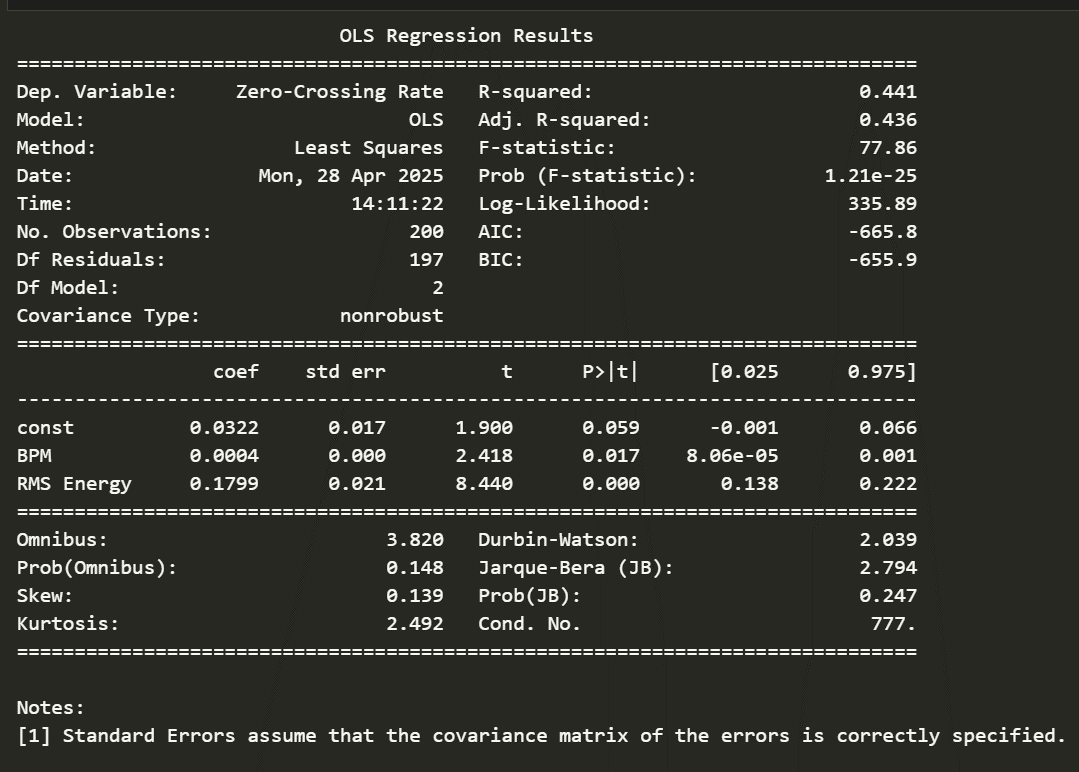


Построим модель чуть-чуть другим способом. Ниже приведен код:

*Рисунок 1.2 — Построение множественной линейной регрессии*

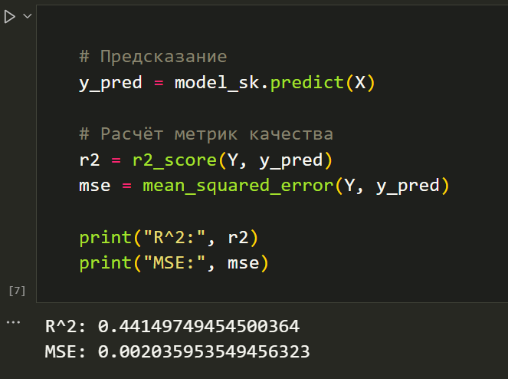


В данном случае мы сразу получаем коэффициенты модели, а рассматривая код 1.1, то вот что он выводит:­

*Рисунок 1.3 — Вывод из кода 1.1.*

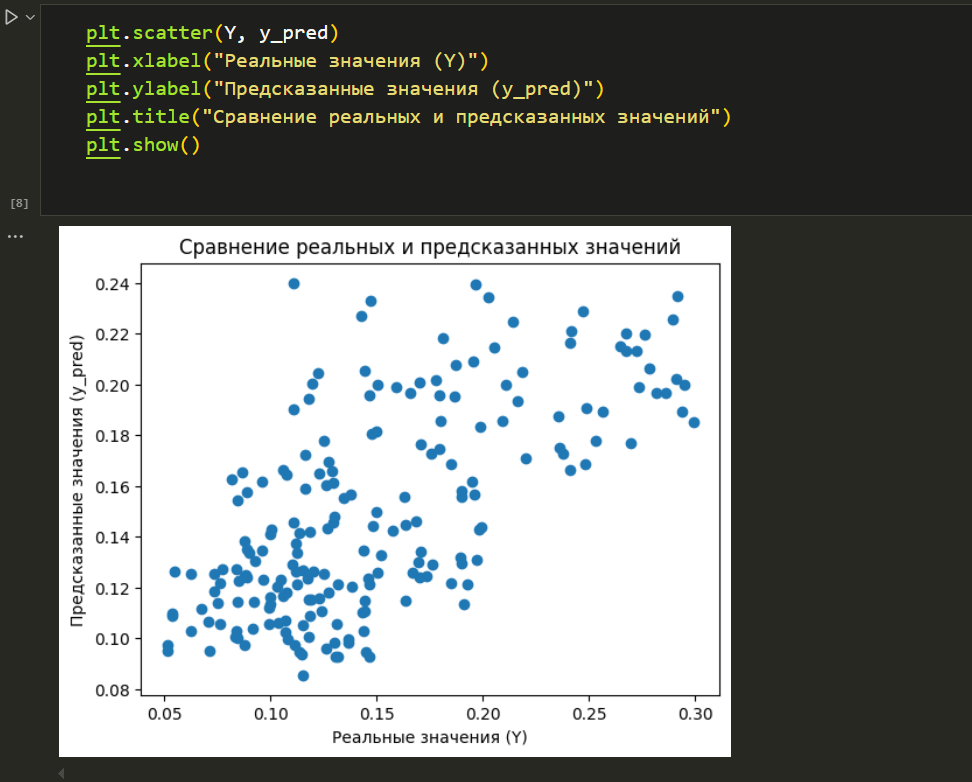
Далее стоит рассчитать такие метрические показатели, как R^2 и MSE, то есть коэффициент детерминации и среднеквадратичную ошибку. Код для расчета:

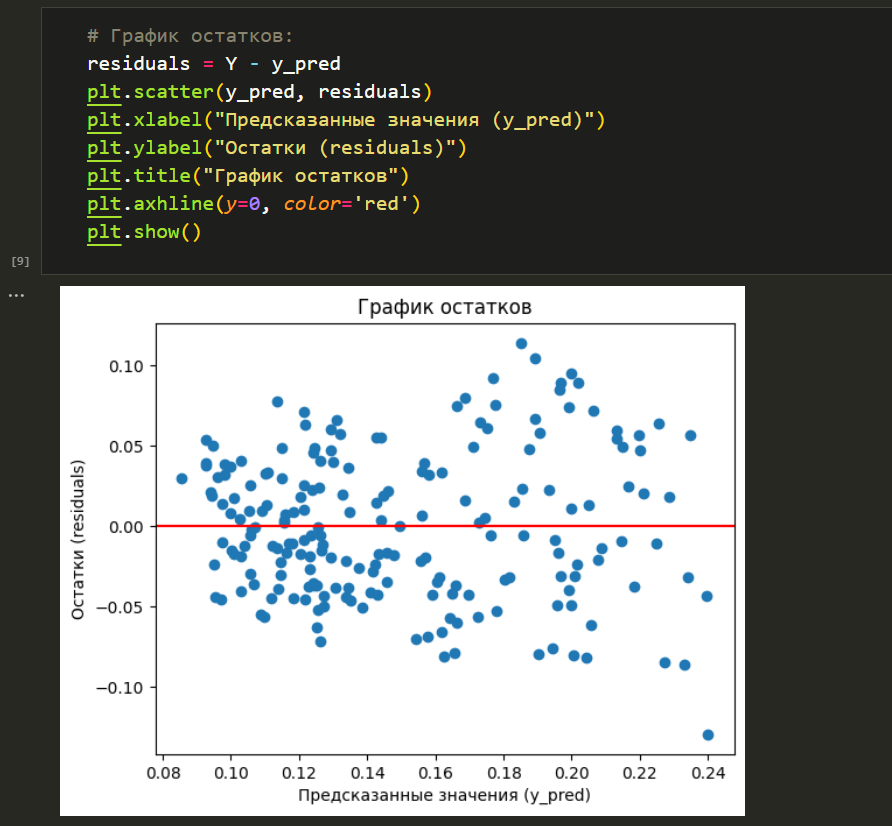
*Рисунок 1.4 — Код для MSE & R^2*



Далее по практической работе нужно построить два графика — это график остатков и график, благодаря которому можно сравнивать реальные и предсказанные значения:

*Рисунок 1.4 — График сравнений.*

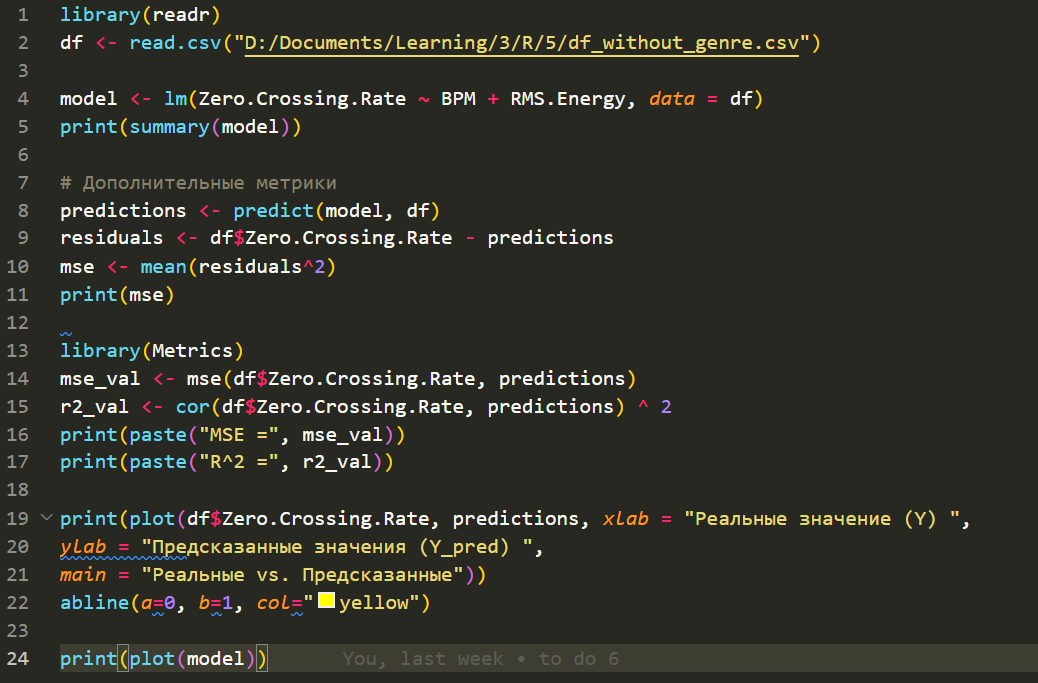


*Рисунок 1.5 — График остатков.*

**Шаг 2) Множественная линейная регрессия в R.**

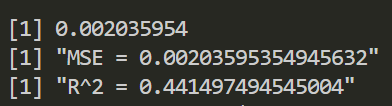
**2.1) Построение множественной линейной регрессии.**

Далее проделаем все то же самое, но с помощью языка программирования R.

*Рисунок 2.1 – Код на языке R.*

В данном коде мы написали сначала код множественной линейной регрессии, потом код для расчета предсказаний, остатков, MSE & R^2.

*Рисунок 2.2 – Метрические показатели.*

****

**3 СРАВНЕНИЕ РЕЗУЛЬТАТОВ**

И там, и там результаты получились одинаковые. Понятное дело, с чуть-чуть разным округлением. Работа в Python куда удобнее чем в R. Если мы рассмотрим график сравнения и остатков, то можно сделать такие выводы:

1) Модель в целом улавливает общую тенденцию, но имеет проблемы с точностью предсказаний.

2) Данные имеют нелинейную структуру, и модель не может адекватно её учитывать.

3) Есть две разные группы данных, которые модель обрабатывает по-разному.

**4 ВЫВОДЫ**

В 6 практической работе приведены реализации множественной линейной регрессии и вычислением метрик регрессии. Также построены графики сравнений и остатков на двух языках программирования.