|  |
| --- |
| 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 №10**

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:**

Изучить основы машинного обучения, построить простые модели в Python, R, а также освоить методы валидации и анализа ошибок.

**Задачи практической работы:**

1 . Построить простые модели машинного обучения:

• Python: использование sklearn для создания моделей классификации и регрессии.

• R: применение пакета caret для построения моделей.

2. Провести валидацию моделей:

• Разделение данных на тренировочную и тестовую выборки.

• Оценка качества моделей на тестовых данных.

• Python: метрики accuracy, mean\_squared\_error.

• R: аналогичные метрики через функции пакета caret.

3. Проанализировать ошибки моделей:

• Построение матрицы ошибок и графиков для анализа ошибок.

4. Сравнить результаты моделей между Python, R.

5. Выявить основные проблемы и сильные стороны инструментов для

задач машинного обучения.

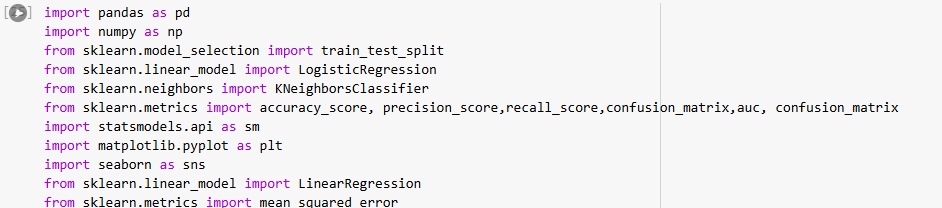
## 2) РЕЗУЛЬТАТЫ ПРАКТИКИ

**Шаг 1) Модели машинного обучения.**

### 1.1) Построение моделей машинного обучения в Python

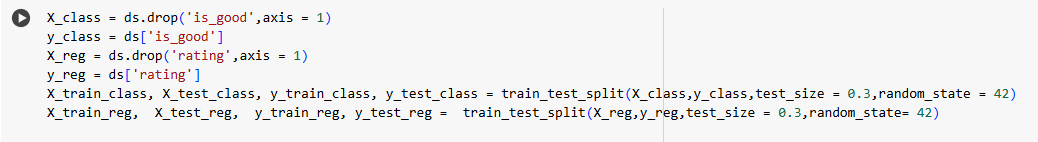
Для начала работы необходимо установить все необходимые библиотеки, так как их достаточно много, то ниже написан код, который показывает все библиотеки, нужные для практической работы.

*Рисунок 1.1 — Необходимые библиотеки.*



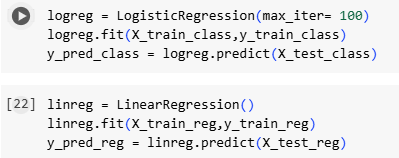
Далее напишем код, с помощью которого мы разделим признаки и целевую для регрессии и классификация, а затем раздели переменные на train и test, для их реализации:

*Рисунок 1.2 – Разделение переменных.*

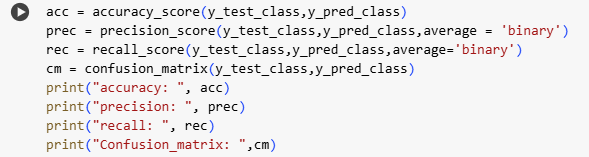


Далее напишем код, с помощью которого будут реализованы логистическая и линейная регресс, а затем посчитаем метрики.

*Рисунок 1.3 — Код логистической и линейной регрессии.*

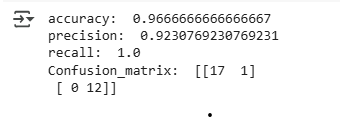


*Рисунок 1.4 - Код метрик.*



Далее посмотрим, какой результат выводит этот код:

*Рисунок 1.5- Метрики.*

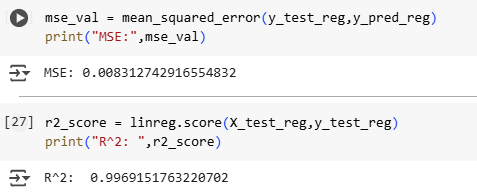
****

Проанализируем это позже в 3 параграфе «сравнение результатов».

Далее напишем код, чтобы рассчитать коэффициент детерминации и

и среднеквадратичную ошибку. Собственно, R^2 и MSE.

*Рисунок 1.6 MSE и R^2.*

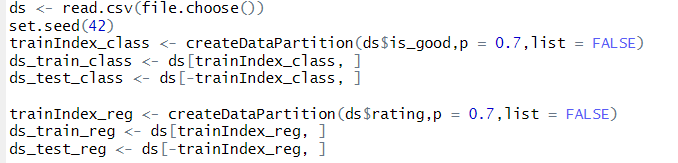
****

**Шаг 2) Модели машинного обучение в R.**

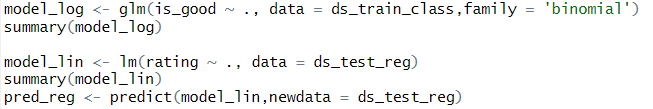
**2.1) Построение простейших моделей машинного обучения в Rstuido.**

Далее проделаем все то же самое, но уже с помощью языка программирования R, убедившись, что все необходимые пакеты были успешно установлены.

*Рисунок 2.1 — Разделение переменных.*

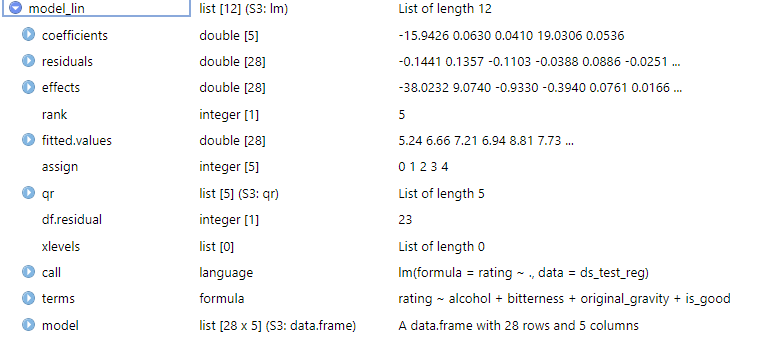
****

*Рисунок 2.2 — Логистическая и линейная регрессия.*

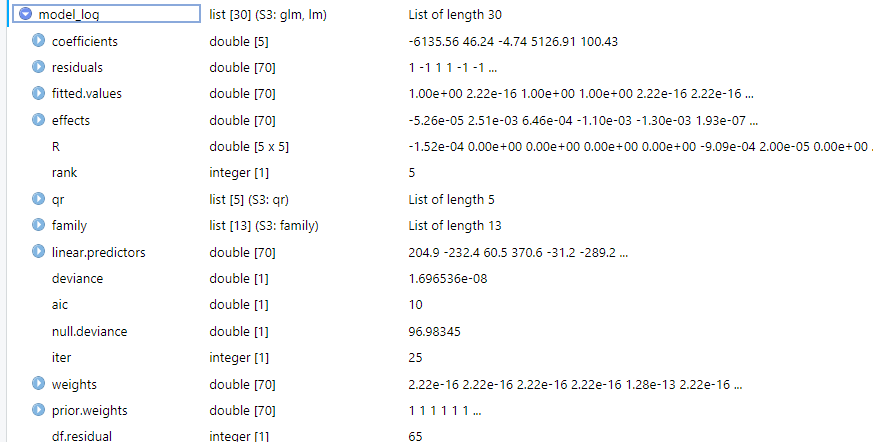
****

На данном этапе мы написали код, для построения обычной логистической модели, а также вычисляет матрицу конфузий и точность.

*Рисунок 2.3 —Результаты линейной регрессии.*

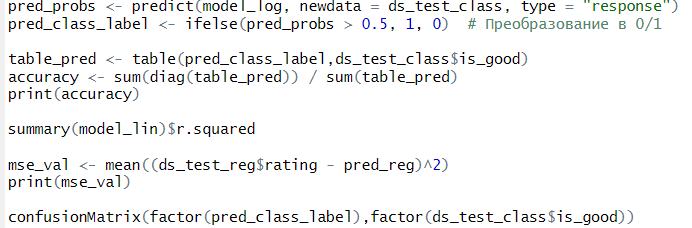
****

*Рисунок 2.4 — Результаты логистической регрессии.*

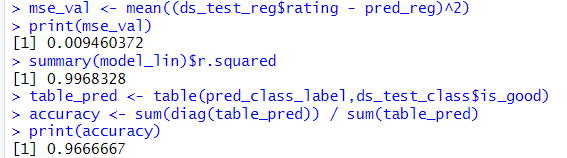
****

Далее рассчитаем все те же коэффициенты:

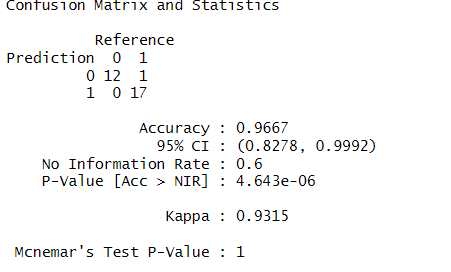
*Рисунок 2.5— Расчет метрик.*

****

*Рисунок 2.6— Вывод метрик.*

****

*Рисунок 2.7— Матрица конфузий.*

****

**3 СРАВНЕНИЕ РЕЗУЛЬТАТОВ**

И там, и там результаты получились одинаковые. Понятное дело работа в R куда удобнее чем в Python, по причине того, что в принципе код куда более компактный, и понятный, единственное, что графики в Python выглядят более красиво, нежели в R.

**MSE (среднеквадратичная ошибка):** MSE на тестовых данных составляет 0.00946. Это говорит о том, что в среднем предсказания модели отклоняются от фактических значений на небольшую величину. Чем меньше MSE, тем лучше модель соответствует данным.

**R-squared (коэффициент детерминации):** R-squared равен 0.9968. Это означает, что 99.68% дисперсии целевой переменной объясняется моделью. Это очень высокий показатель, что говорит об отличной подгонке модели к данным. Однако, как и в предыдущем случае, такой высокий R-squared может указывать на переобучение.

**Линейная регрессия:** Модель линейной регрессии показывает очень хорошие результаты (низкий MSE, высокий R-squared). Однако, необходимо проверить, нет ли переобучения.

**Классификация:** Модель классификации показывает высокую точность. Но важно оценить и другие метрики (precision, recall, F1-score), особенно если классы несбалансированы.

**4 ВЫВОДЫ**

В результате 10 практической работы, мы ознакомились со способами реализации линейной регрессии и логистической регрессии. Разобрались в простейших моделях машинного обучения и посчитали самые базовые коэффициенты, перед этим разделив переменные для регрессии и классификации.