|  |
| --- |
| 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 №9**

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 классификации (логистическая регрессия и KNN) с использованием Python, R, а также интерпретацию результатов классификации через отчёты.

**Задачи практической работы:**

1. Построить модель логистической регрессии:

• Python: использование библиотеки sklearn и statsmodels.

• R: функция glm() с семейством binomial.

2. Построить модель K-ближайших соседей (KNN):

• Python: библиотека sklearn (KneighborsClassifier).

• R: использование пакета class или caret.

3. Выполнить прогноз на тестовых данных:

• Python и R: метрики оценки точности (accuracy, precision, recall). 4. Интерпретировать результаты моделей:

• Выявить важные переменные в логистической регрессии.

• Оценить точность моделей с помощью ROC-кривой, матрицы ошибок.

• Сравнить результаты между Python, R

## 2) РЕЗУЛЬТАТЫ ПРАКТИКИ

**Шаг 1) Knn и логистические регрессии.**

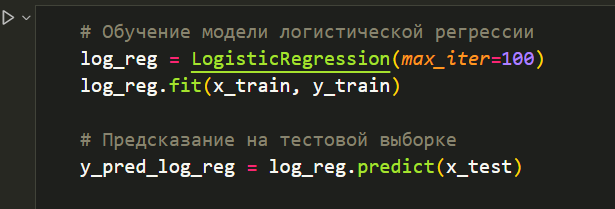
### 1.1) Knn и логистические регрессии в Python.

Для более корректной оценки качества классификации необходимо иметь две части данных: Train (для обучения), Test (для проверки качества). Проведем разбиение признаков и целевой переменной. Ниже представлен код:

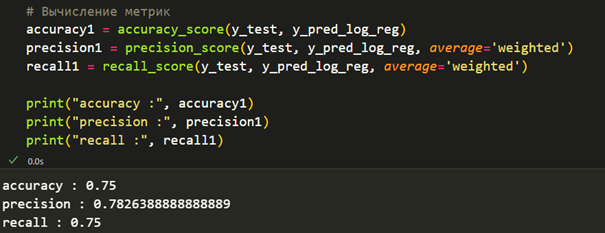
*Рисунок 1.1 — Разделение на train/test.*

Далее напишем код, который реализует логистическую регрессию. Сначала с помощью библиотеки sklearn, а замет statsmodels. Ниже представлен код sklear логистической регрессии:

*Рисунок 1.2 – логистическая регрессия sklearn.*

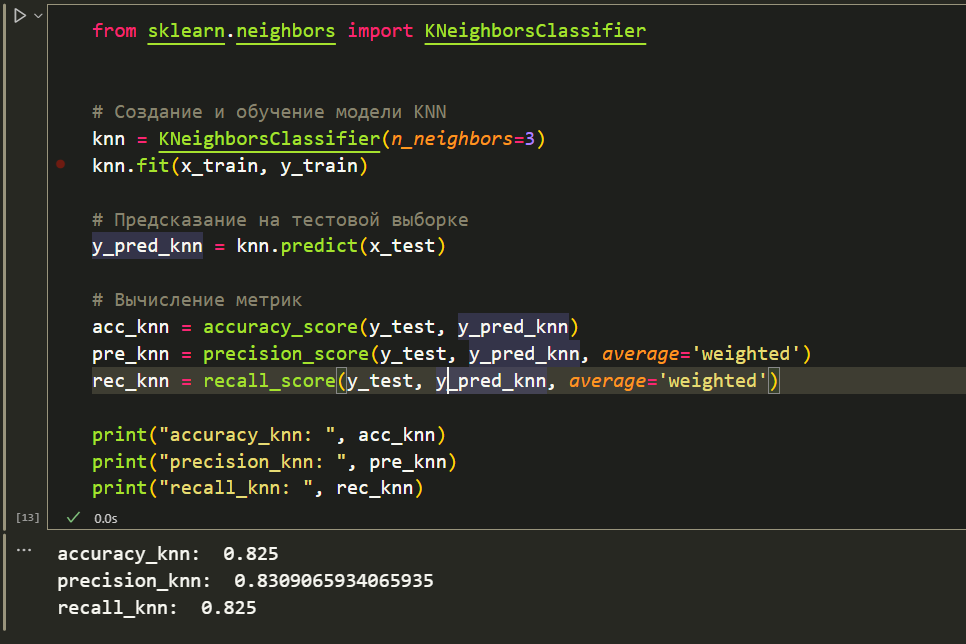


Дальше вычислим метрики логистической регрессии, для изучения ее метрики оценивания: accuracy, recall и precision:

*Рисунок 1.3 — Метрики.*

Вот какой результат выводит этот код. Можно заметить, что регрессия получилось не очень точной, возможно, из-за того, что для обучения модели требуется больше данных.

Далее построим KNN (метод ближайшего соседа) и вычислим все те же метрики, что делали до этого.

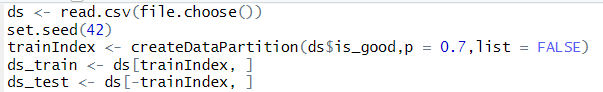
*****Рисунок 1.6 KNN регрессия и метрики.*

**Шаг 2) Knn и логистическая регрессия в R.**

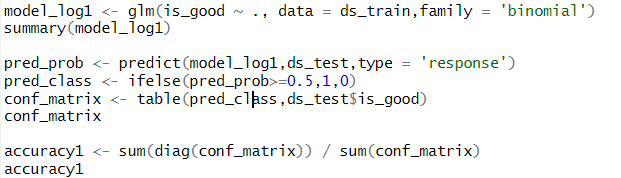
**2.1) Knn и логистическая регрессия в R.**

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 на train/test.*

****

*Рисунок 2.2 — Логистическая регрессия и метрики*

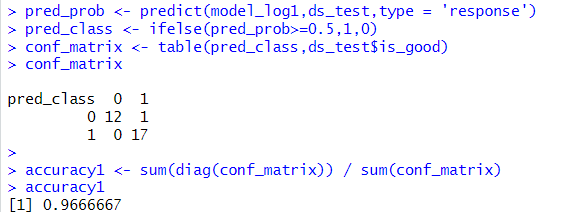
****

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

*Рисунок 2.3 — Показатели регрессии.*

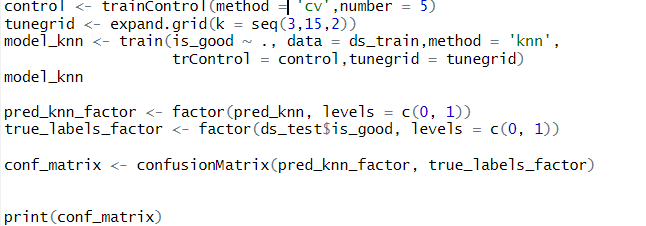
****

*Рисунок 2.4 — Матрица и точность.*

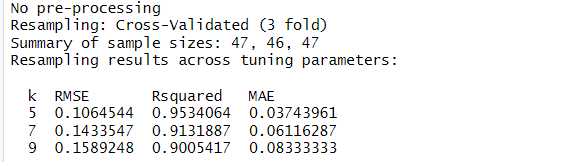
****

Далее построим KNN регрессию.

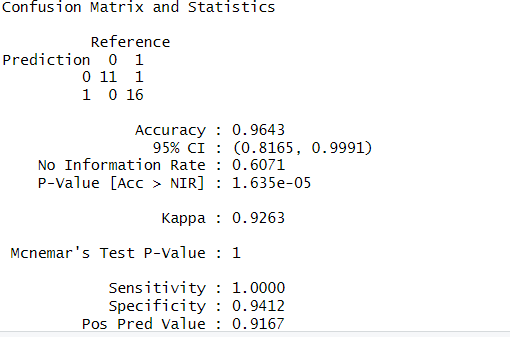
*Рисунок 2.5— KNN регрессия.*

****

*Рисунок 2.6— Вывод.*

****

*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.

P-значения для всех предикторов (кроме rating) очень близки к 1, что говорит о том, что эти переменные, вероятно, не оказывают значительного влияния на целевую переменную. Это может быть связано с небольшим размером выборки или мультиколлинеарностью (зависимостью между предикторами).

AUC (Area Under the Curve) = 1.00. Это идеальный результат, что означает идеальную классификацию. Как и с R-squared, это может указывать на переобучение или другую проблему с данными/моделью. В реальных задачах AUC обычно ниже.

**4 ВЫВОДЫ**

В результате 9 практической работы, мы ознакомились со способами реализации KNN регрессии и логистической регрессии. Разобрались, в какой среле разработки более удобно работать, а также подвели общие итоги относительно кластеризации.