Инженерный вестник Дона, №8 (2024)
ivdon.ru/ru/magazine/archive/n8y2024/9414

Разработка web-приложения для предобработки данных с использованием библиотек Python

Д.С. Пономарев

Научно-исследовательский институт Федеральной службы исполнения наказаний, г. Москва
Ижевский государственный технический университет имени М.Т. Калашникова, г. Ижевск

Аннотация: В статье рассмотрена разработка инструментов нормализации и стандартизации данных при помощи библиотек Python. Рассмотрено описание теоретических основ и формул, используемых для нормализации и стандартизации данных. Для внутренних расчетов разрабатываемого программного обеспечения были использованы библиотеки Pandas, NumPy. Внешний интерфейс был построен на основе библиотеки Streamlit, которая позволяет развертывать web-приложения без каких-либо дополнительных ресурсов. Приведены фрагменты кода, объяснены механизмы реализации. Проведено описание разработанного инструмента: подробное объяснение функциональности инструмента, интерфейса пользователя и примеров использования. Рассмотрена важность предварительной обработки данных, выбор подходящего метода и заключительные замечания о пользе интерактивных инструментов для обработки данных.
Ключевые слова: обработка данных, статистика, информационные системы, web-системы *Python*.

Введение

В современном мире данных, растущие объемы информации требуют эффективных методов их обработки и анализа. Важными этапами информационных систем и разрабатываемого программного обеспечения являются проработанная архитектура базы данных [1], использование современных методов интеллектуальной поддержки принятия решений [2], разработка инструментов анализа и грамотной предобработки информации [3]. Нормализация и стандартизация данных являются ключевыми этапами предварительной обработки данных [4], обеспечивающими приведение различных переменных к единому масштабу. Эти методы используются в различных областях, включая машинное обучение, статистику и бизнес-аналитику, где корректная подготовка данных напрямую влияет на точность и надежность моделей и выводов.

Однако выбор метода нормализации или стандартизации данных не всегда является очевидным и может зависеть от множества факторов, включая природу данных и цель анализа. В статье представлен интерактивный инструмент, разработанный на Python с использованием библиотек Pandas [5], NumPy [6], Streamlit [7], который позволяет пользователям загружать данные, выбирать методы нормализации и стандартизации, а также сохранять обработанные данные для дальнейшего использования.

Теоретические основы и разработка кода для обработки данных

Рассмотрим пример разработки инструмента для нормализации и стандартизации данных [8].

1. Для нормализации данных были рассмотрены Min-Max нормализация и робастная нормализация.

1.1. Для Min-Max нормализации применена формула (1) [9]:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad (1)$$

где: x — исходное значение; min(x) — минимальное значение в столбце; max(x) — максимальное значение в столбце; x′ — нормализованное значение.

Реализация была сделана следующим образом:

max_value = df [column].max()

min_value = df [column].min()

result[column] = (df[column] - min_value) / (max_value - min_value)

1.2. Робастная нормализация (2) [10]:

$$x' = \frac{x - Q_1}{Q_3 - Q_1}, \quad (2)$$

где: x — исходное значение; Q1 — первый квартиль (25-й процентиль); Q3 — третий квартиль (75-й процентиль); Q3−Q1 — межквартильный диапазон (IQR); x′ — нормализованное значение.

Реализация в коде:

q75, q25 = np.percentile(df[column], [75 ,25])

iqr = q75 - q25

result[column] = (df[column] - q25) / iqr

2. Для стандартизации были рассмотрены Z-score и стандартизация по медиане и межквартильному диапазону.

2.1. Z-score стандартизация (3) [9, 10]:

$$z = \frac{x - \mu}{\sigma}, \quad (3)$$

где: x — исходное значение; μ — среднее значение в столбце; σ — стандартное отклонение в столбце; z — стандартизированное значение.

Реализовано следующим образом:

mean = df[column].mean()

std = df[column].std()

result[column] = (df[column] - mean) / std

2.2. Стандартизация по медиане и межквартильному диапазону (4) (скейлинг) [10]:

$$x' = \frac{x - median(x)}{Q_3 - Q_1}, \quad (4)$$

где: median(x) — медианное значение.

Реализация:

median = df[column].median()

q75, q25 = np.percentile(df[column], [75 ,25])

iqr = q75 - q25

result[column] = (df[column] - median) / iqr

Выбор подходов для нормализации данных был записан в функцию «normalize_data»; стандартизации данных – в функцию «standardize_data». Данные функции принимают две переменные – данные («df») и выбранный пользователем метод («method»). Для выбора конкретного метода внутри

функций использовалась конструкция «if-elif». На рисунке далее представлен пример функции для выбора метода нормализации данных (рисунок 1).

```
def normalize_data(df, method):
    result = df.copy()
    for column in df.select_dtypes(include=[np.number]).columns:
        if method == "Min-Max":
            max_value = df[column].max()
            min_value = df[column].min()
            result[column] = (df[column] - min_value) / (max_value - min_value)
        elif method == "Робастная нормализация":
            q75, q25 = np.percentile(df[column], [75 ,25])
            iqr = q75 - q25
            result[column] = (df[column] - q25) / iqr
    return result
```

Рис 1. – Фрагмент кода для обработки данных на примере рассмотренных методов нормализации

Аналогичным образом строится и функция «standardize_data» для выбора метода стандартизации исследуемых данных.

Создание интерфейсов и примеры использования

Рассмотрим разработку пользовательских интерфейсов для выбора определенного метода. Для этого был использован метод «st.selectbox». Также был рассмотрен случай, когда нужна работа с фактическими данными без применения стандартизации и нормализации данных. Разработка представлена далее на рисунке 2.

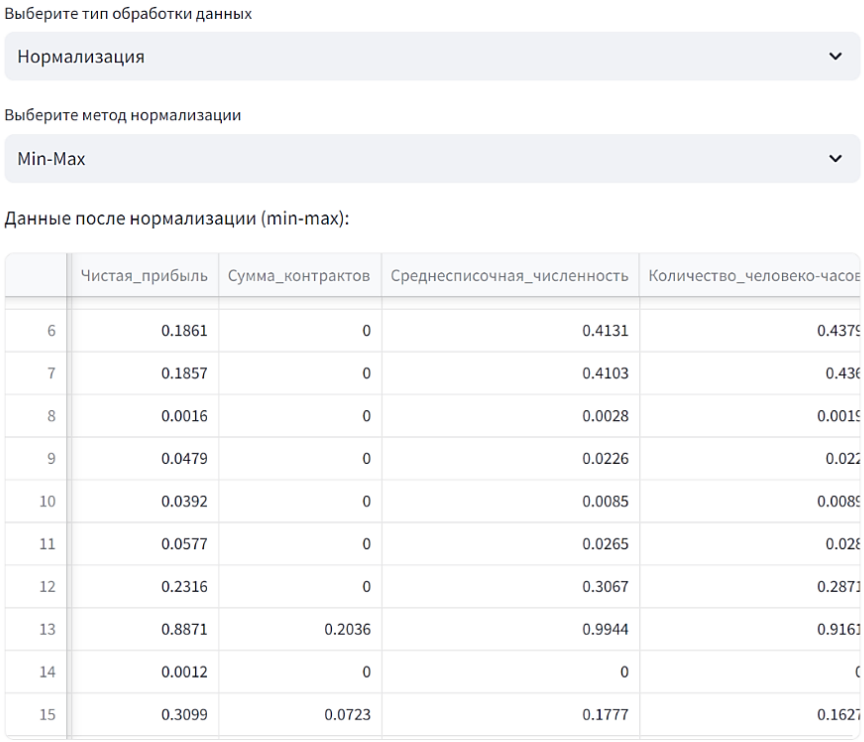
```
process_type = st.selectbox("Выберите тип обработки данных", ("Без обработки", "Нормализация", "Стандартизация"))

if process_type == "Нормализация":
    method = st.selectbox("Выберите метод нормализации", ("Min-Max", "Робастная нормализация"))
    df_processed = normalize_data(df, method)
    st.write(f"Данные после нормализации ({method.lower()}):")
    st.write(df_processed)
elif process_type == "Стандартизация":
    method = st.selectbox("Выберите метод стандартизации", ("Z-score", "Скейлинг по медиане и межквартильному диапазону"))
    df_processed = standardize_data(df, method)
    st.write(f"Данные после стандартизации ({method.lower()}):")
    st.write(df_processed)
else:
    df_processed = df
    st.write("Данные без обработки:")
    st.write(df_processed)
```

Рис. 2. – Разработка кода для реализации пользовательских интерфейсов выбора методов стандартизации или нормализации

Рассмотрим более подробно элементы данного кода. Как видно, при помощи связки из «if-elif-else» пользователю предлагается сделать выбор одного из типов обработки данных, а также методов в рамках уже выбранного типа. Для непосредственно самих расчетов использованы функции «normalize_data» и «standardize_data», которые были разобраны ранее.

В методе «st.write» используются f-строки для заголовка, чтоб обозначить какой тип и метод в итоге были выбраны (а также был применен метод lower (), чтоб запись была более корректно отражена). Также при помощи «st.write» визуализируется обработанная выборка (т.е. «st.write(df_processed)»). В результате был получен следующий интерфейс (рисунок 3).



	Чистая_прибыль	Сумма_контрактов	Среднесписочная_численность	Количество_человеко-часов
6	0.1861	0	0.4131	0.4379
7	0.1857	0	0.4103	0.436
8	0.0016	0	0.0028	0.0019
9	0.0479	0	0.0226	0.022
10	0.0392	0	0.0085	0.0089
11	0.0577	0	0.0265	0.028
12	0.2316	0	0.3067	0.2871
13	0.8871	0.2036	0.9944	0.9161
14	0.0012	0	0	0
15	0.3099	0.0723	0.1777	0.1627

Рис. 3. – Итоговое окно для проведения стандартизации и нормализации данных, визуализации полученной выборки и ее сохранения

Случаи, когда может быть использован разработанный инструмент обработки данных: Min-Max Нормализация: если есть значения в диапазоне от 10 до 20, после нормализации они будут в диапазоне от 0 до 1; робастная нормализация: если набор данных имеет выбросы, робастная нормализация уменьшит их влияние, используя медиану и межквартильный диапазон; Z-score стандартизация: преобразует значения так, что они будут иметь среднее значение 0 и стандартное отклонение 1; стандартизация по медиане и межквартильному диапазону: подходит для данных с выбросами, так как медиана и межквартильный диапазон менее подвержены влиянию выбросов, чем среднее и стандартное отклонение. Использование этих методов помогает подготовить данные для дальнейшего анализа, уменьшая влияние масштаба и выбросов.

Заключение

В статье приведены примеры использования популярных библиотек Python для обработки данных и последующего развертывания в сети интернет. Разработанная web-система представляет собой достаточно удобное средство для предварительной обработки данных, облегчая задачу выбора и применения различных методов нормализации и стандартизации. Разработанная система будет особенно полезна для аналитиков данных, исследователей и практиков машинного обучения, предоставляя интуитивно понятный интерфейс и гибкие возможности для обработки данных.

Литература

1. Мартынов В.А., Плотникова Н.П. Применение дерева отрезков в PostgreSQL // Инженерный вестник Дона, 2023, № 9. URL: ivdon.ru/ru/magazine/archive/n9y2023/8684

2. Передерий В.А., Рысин М.Л. Сравнительная характеристика библиотек машинного обучения для внедрения искусственного интеллекта в CRM-систему // Инженерный вестник Дона, 2024, № 4. URL: ivdon.ru/ru/magazine/archive/n4y2024/9160

3. Благодатский Г.А., Горохов М.М. Пономарев С.Б. Системный анализ организационной структуры медицинской службы уголовно-исполнительной системы и управление ее реформированием/Изд-во ИжГТУ им. М.Т. Калашникова, 2017. 104 с.

4. Горохов М.М., Пономарев С.Б. Статистические методы анализа и обработки информации: нейронные сети //Научные труды ФКУ НИИ ФСИН России, 2018. № 2. С. 140-146.

5. Harrison M. Effective Pandas: Patterns for Data Manipulation. Independently published. 2021. 497 p.

6. Idris I. Numpy Beginner's Guide. 2015. Packt Publishing 348 p.

7. Khorasani M., Abdou M., Fernandez J.H. Web Application Development with Streamlit. Apress. 2022. 508 p.

8. Bruce P., Bruce A., Gedeck P. Practical statistics for Data Scientists / O’Reilly, 2020, pp. 1-86.

9. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction / New York: Springer, 2001, pp. 50-165.

10. Rousseeuw Peter J., Ronchetti Elvezio M. Robust Statistics / Wiley, 2019, pp. 121-209.

References

1. Marty`nov V.A., Plotnikova N.P. Inzhenernyj vestnik Dona, 2023, № 9. URL: ivdon.ru/ru/magazine/archive/n9y2023/8684

2. Perederij V.A., Ry`sin M.L. Inzhenernyj vestnik Dona, 2024, № 4. URL: ivdon.ru/ru/magazine/archive/n4y2024/9160

3. Blagodatskij G.A., Gorokhov M.M. Ponomarev S.B. Sistemny`j analiz organizacionnoj struktury` medicinskoj sluzhby` ugolovno-ispolnitel`noj sistemy` i upravlenie ee reformirovaniem [Systematic analysis of the organizational structure of the medical service of the penal system and the management of its reform]. IzhGTU im. M.T. Kalashnikova, 2017. 104 p.

4. Gorokhov M.M., Ponomarev S.B. Nauchny`e trudy` FKU NII FSIN Rossii, 2018. № 2. S. 140-146.

5. Harrison M. Effective Pandas: Patterns for Data Manipulation. Independently published. 2021. 497 p.

6. Idris I. Numpy Beginner's Guide. 2015. Packt Publishing 348 p.

7. Khorasani M., Abdou M., Fernandez J.H. Web Application Development with Streamlit. Apress. 2022. 508 p.

8. Bruce P., Bruce A., Gedeck P. Practical statistics for Data Scientists. O'Reilly, 2020, pp. 1-86.

9. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer, 2001, pp. 50-165.

10. Rousseeuw Peter J., Ronchetti Elvezio M. Robust Statistics. Wiley, 2019, pp. 121-209.

Дата поступления: 8.06.2024

Дата публикации: 18.07.2024