Пономарев Александр Николаевич, магистрант,
Вятский государственный университет
Ponomarev Aleksandr Nikolaevich,
Vyatka State University

ПРОГРАММНЫЙ КОМПЛЕКС АВТОМАТИЗАЦИИ ОЧИСТКИ ТАБЛИЧНЫХ ДАННЫХ С ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКОЙ
A SOFTWARE COMPLEX FOR AUTOMATING TABULAR DATA CLEANING WITH INTELLIGENT SUPPORT OF PREPROCESSING METHOD SELECTION

Аннотация. Представлен программный комплекс для автоматизации очистки табличных данных средствами языка Python. Описаны архитектура веб-приложения и алгоритм автоматического подбора методов предобработки по результатам разведочного анализа.

Abstract. A software complex for automating tabular data cleaning using Python is presented. The architecture of the web application and the algorithm of automatic selection of preprocessing methods based on exploratory data analysis are described.

Ключевые слова: Предобработка данных, очистка данных, разведочный анализ, автоматизация, Python, программный комплекс.

Keywords: Data preprocessing, data cleaning, exploratory data analysis, automation, Python, software complex.

Подготовка данных является одним из наиболее трудоёмких этапов работы аналитика: по данным отраслевых исследований, на сбор, очистку и преобразование данных приходится от 60 до 80 % рабочего времени специалиста [4]. Систематизация методов предобработки рассматривается в работах [1, 2]. Существующие инструменты экосистемы Python – pandas, scikit-learn [5], ydata-profiling, great-expectations, pandera – ориентированы на программную работу и охватывают отдельные аспекты предобработки. Отсутствует общедоступное решение, объединяющее разведочный анализ, конфигурируемое применение последовательности методов очистки и автоматический подбор шагов в едином веб-интерфейсе. Настоящая работа посвящена разработке такого программного комплекса.

Программный комплекс реализован как клиент-серверное веб-приложение трёхзвенной архитектуры: клиент в виде браузера, сервер приложений на базе фреймворка FastAPI [11] и хранилище из СУБД SQLite и файловой системы. Серверные компоненты упакованы в Docker-контейнеры. Система декомпозирована на тринадцать функциональных подсистем: аутентификации, загрузки и хранения датасетов, разведочного анализа, обработки пропусков, обработки выбросов, масштабирования признаков, кодирования категориальных и текстовых признаков, дедупликации и приведения типов, конфигурируемого pipeline и графического конструктора, пользовательских функций, генерации отчётов, асинхронного выполнения и журналирования событий.

Подсистема загрузки поддерживает форматы CSV, TSV, XLSX, JSON, JSON Lines и Parquet объёмом до 30 ГБ с потоковым приёмом, автоматическим определением кодировки и эвристическим распознаванием формата по первым байтам сигнатуры. Подсистема разведочного анализа [7] формирует сводную описательную статистику: общие характеристики датасета, статистики числовых признаков (среднее, медиана, среднеквадратическое отклонение, асимметрия, эксцесс, квартили), характеристики категориальных колонок, матрицы корреляций Пирсона и Спирмена. Визуализация выполняется на стороне клиента средствами Plotly.js, результаты кэшируются в базе данных.

В подсистемах очистки реализовано сорок два метода, объединённых в пять категорий, сводный состав которых приведён в таблице 1.

Таблица 1

Состав реализованных методов очистки данных

Категория	Количество	Реализованные методы
Обработка пропусков	13	удаление строк и колонок; распознавание неявных пропусков; заполнение средним, медианой, модой, константой, специальной категорией; импутация методом k ближайших соседей; итеративная импутация MICE; forward-fill, backward-fill; линейная интерполяция
Обработка выбросов	5	правило трёх сигм по z-критерию; правило межквартильного размаха; усечение по процентилям (winsorize); Isolation Forest; Local Outlier Factor
Масштабирование и нормализация	8	стандартизация; минимаксная нормализация; робастная нормализация; масштабирование по модулю максимума; логарифмическое преобразование; преобразование Бокса – Кокса; преобразование Йео – Джонсона; квантильное преобразование
Кодирование категориальных и текстовых признаков	11	Label Encoding; One-Hot Encoding; Frequency Encoding; Target Encoding с байесовским сглаживанием; Hash Encoding; объединение редких категорий; приведение к нижнему регистру; удаление лишних пробелов; нормализация Unicode; замена по регулярному выражению; извлечение частей даты
Дедупликация и приведение типов	5	удаление точных дубликатов; нечёткая дедупликация по Левенштейну и Jaro – Winkler; автоматическое определение типов; нормализация имён колонок; парсинг дат
Итого	42	–

Базовый класс шага очистки имеет единый интерфейс с методом fit_transform, что обеспечивает единообразие использования и сериализацию конфигурации в формате YAML.

Среди числовых методов наиболее существенными являются итеративная многомерная импутация пропусков MICE [8], при которой каждый признак восстанавливается регрессией по остальным до сходимости; обнаружение выбросов алгоритмом Isolation Forest [9] и методом локального фактора выброса LOF [10]; преобразования Бокса – Кокса [12] и Йео – Джонсона, параметр которых оценивается методом максимального правдоподобия из условия приближения преобразованного распределения к нормальному. Для категориальных признаков, помимо классических Label и One-Hot Encoding, реализовано Target Encoding с байесовским сглаживанием [3], регуляризующее условное среднее целевой переменной по категории и предотвращающее переобучение на редких значениях. Нечёткая дедупликация записей опирается на меры сходства строк – расстояние Левенштейна [13] и меру Jaro – Winkler.

Последовательность шагов очистки описывается как упорядоченный список конфигураций, сериализуемый в формат YAML, что обеспечивает воспроизводимость и переносимость pipeline между датасетами. Графический конструктор в браузере позволяет добавлять, удалять, переставлять шаги и редактировать их параметры без программирования. Применение pipeline возможно в синхронном и в асинхронном режиме; последний реализован через пул потоков с отображением прогресса и возможностью отмены задачи пользователем.

Главной отличительной особенностью разработанного программного комплекса является подсистема автоматического подбора рекомендуемых шагов очистки. Алгоритм работает над разведочным отчётом и формирует упорядоченный список рекомендаций, каждая из которых содержит готовую конфигурацию шага, текстовое обоснование на естественном языке и метку приоритета – высокий, средний или низкий. Сводный перечень одиннадцати реализованных правил с условиями срабатывания и рекомендуемыми методами приведён в таблице 2.

Таблица 2

Правила автоматического подбора методов очистки

№	Условие срабатывания	Рекомендуемый метод	Приоритет
1	Наличие нестандартных имён колонок (заглавные, пробелы, спецсимволы)	Нормализация имён колонок к snake_case	Средний
2	Доля полных дубликатов строк больше нуля	Удаление точных дубликатов	Высокий
3	Доля пропусков в колонке превышает 60 %	Удаление колонки	Высокий
4	Числовая колонка с пропусками, |skewness| > 1	Заполнение медианой	Высокий
5	Числовая колонка с пропусками, |skewness| ≤ 1	Заполнение средним арифметическим	Высокий
6	Категориальная или текстовая колонка с пропусками	Заполнение специальной категорией «MISSING»	Высокий
7	Доля выбросов по IQR в колонке превышает 5 %	Усечение значений по правилу межквартильного размаха (clip)	Средний
8	Категориальная колонка с кардинальностью 2–10 значений	One-Hot Encoding	Средний
9	Категориальная колонка с кардинальностью свыше 10 значений	Label Encoding или Frequency Encoding	Средний
10	Числовая колонка с |skewness| > 2	Преобразование Йео – Джонсона	Низкий
11	Отношение размахов числовых колонок превышает 100×	Стандартизация (StandardScaler)	Низкий

Рекомендации возвращаются пользователю в виде карточек с пояснением. Пользователь может принять все рекомендации единым действием, выбрать отдельные либо отредактировать параметры перед запуском. При коллективном принятии шаги автоматически упорядочиваются в логически корректной последовательности: дедупликация, удаление

неинформативных колонок, обработка пропусков и выбросов, кодирование категорий, нормализация распределений, приведение к единому масштабу. Подсистема пользовательских функций обеспечивает расширение каталога методов собственными функциями на языке Python без модификации исходного кода. Перед сохранением выполняется статический анализ кода средствами модуля ast: проверяется отсутствие запрещённых имён, импортов вне разрешённого перечня модулей и обращений к служебным атрибутам объектов. Исполнение производится в ограниченном пространстве имён с тайм-аутом тридцать секунд.

Технологической основой описанного программного комплекса служит язык Python версии 3.11. Серверная часть построена на асинхронном веб-фреймворке FastAPI [11], валидация входных и выходных данных обеспечивается библиотекой Pydantic второй версии, работа с базой данных осуществляется через ORM SQLAlchemy 2.0 с системой миграций Alembic. Численная обработка табличных данных опирается на Pandas [6, 15] и NumPy; для файлов объёмом свыше одного гигабайта применяется колоночная библиотека Polars. Статистические вычисления выполнены средствами SciPy, преобразования признаков и реализации методов масштабирования – средствами scikit-learn [5], автоматическое профилирование датасета – средствами ydata-profiling. Веб-интерфейс выполнен на шаблонизаторе Jinja2 с применением технологии HTMX и CSS-фреймворка Tailwind CSS; интерактивная визуализация – средствами Plotly.js. Генерация PDF-отчётов реализована через WeasyPrint. Контейнеризация серверных компонентов выполнена средствами Docker Compose. Качество кодовой базы поддерживается за счёт модульного тестирования с использованием pytest и статического анализа линтерами ruff и mypy.

Разработанный программный комплекс объединяет в едином веб-интерфейсе функции разведочного анализа, конфигурируемого применения методов очистки, расширения пользовательскими функциями и формирования отчётов. Подсистема автоматического подбора методов снижает порог входа [14] для специалиста и одновременно сохраняет гибкость за счёт возможности ручной корректировки рекомендаций. Перспективные направления развития – расширение базы правил подбора, интеграция с фреймворками автоматизированного машинного обучения и поддержка обработки потоковых данных.

Список литературы:

1. Wickham H. Tidy Data // Journal of Statistical Software. – 2014. – Vol. 59, № 10. – P. 1–23.
2. García S., Luengo J., Herrera F. Data Preprocessing in Data Mining. – Cham: Springer International Publishing, 2015. – 320 p.
3. Kuhn M., Johnson K. Feature Engineering and Selection: A Practical Approach for Predictive Models. – Boca Raton: CRC Press, 2019. – 297 p.
4. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter. – 3rd ed. – Sebastopol: O'Reilly Media, 2022. – 581 p.
5. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2825–2830.
6. McKinney W. Data Structures for Statistical Computing in Python // Proceedings of the 9th Python in Science Conference. – 2010. – P. 56–61.
7. Tukey J.W. Exploratory Data Analysis. – Reading: Addison-Wesley Publishing Company, 1977. – 688 p.
8. Van Buuren S. Flexible Imputation of Missing Data. – 2nd ed. – Boca Raton: Chapman and Hall / CRC, 2018. – 444 p.
9. Liu F.T., Ting K.M., Zhou Z.-H. Isolation Forest // Proceedings of the 8th IEEE International Conference on Data Mining. – Pisa: IEEE, 2008. – P. 413–422.

10. Breunig M.M., Kriegel H.-P., Ng R.T., Sander J. LOF: Identifying Density-Based Local Outliers // ACM SIGMOD Record. – 2000. – Vol. 29, № 2. – P. 93–104.
11. Documentation of FastAPI. – [Электронный ресурс]. – URL: https://fastapi.tiangolo.com/ (дата обращения: 28.04.2026).
12. Box G.E.P., Cox D.R. An Analysis of Transformations // Journal of the Royal Statistical Society. Series B (Methodological). – 1964. – Vol. 26, № 2. – P. 211–252.
13. Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академии наук СССР. – 1965. – Т. 163, № 4. – С. 845–848.
14. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. – 3rd ed. – Waltham: Morgan Kaufmann, 2012. – 740 p.
15. Документация библиотеки Pandas. – [Электронный ресурс]. – URL: https://pandas.pydata.org/docs/ (дата обращения: 08.05.2026).