САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики-процессов управления

Программа бакалавриата
“Большие данные и распределенная цифровая платформа”

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Алгоритмы и структуры данных»

на тему «Метода заполнения пропусков»

Студент гр. 23Б15-пу
Трофимов И. А.

Преподаватель
Дик А.Г.

Санкт-Петербург

2025 г.

Оглавление

Цель работы

Целью данной лабораторной работы является разработка и реализация программного средства для восстановления пропущенных значений в табличных данных с использованием трёх методов — Удаление строк с NaN, повторение результата последнего наблюдения перед пропуском и заполнение пропусков на основе линейной регрессии — а также проведение их экспериментального сравнения по метрикам суммарной относительной ошибки и доли некорректных восстановлений на синтетических датасетах различных размеров и уровней пропусков.

Теоретическая часть

В процессе сбора и обработки данных практически всегда возникает проблема пропусков значений (missing values). Пропуски могут появляться по разным причинам: технические сбои, ошибки ввода, отказ пациента от сдачи анализа и т.д. Пропуски данных существенно затрудняют проведение статистического анализа, а иногда делают невозможным использование ряда аналитических методов, требующих полной информации.

Основные причины появления пропусков:

- Технические ошибки при сборе данных (сбой оборудования, потеря связи).
- Нарушение процедуры ввода или переноса данных.
- Человеческий фактор: забывчивость, отказ от предоставления информации.
- Экспериментальный дизайн (отсутствие необходимости измерения какого-либо параметра для части наблюдений).

Методы обработки пропусков

Существует несколько подходов к работе с пропущенными значениями:

1. Удаление строк с пропусками (complete case analysis)

   Простой, но зачастую неэффективный метод. При большом количестве пропусков может сильно уменьшить объем данных и исказить результаты, если пропуски не случайны

2. Заполнение пропусков методом последнего наблюдения (Last Observation Carried Forward, LOCF):

   Пропуск заменяется последним известным значением признака. Эффективно при временных рядах и когда значения показателя меняются плавно

3. Заполнение пропусков на основе регрессионной модели (linear regression imputation):

   Для заполнения пропусков используется модель, построенная на основе других наблюдений. Более сложный, но и более точный метод, особенно при наличии сильной связи между признаками.

Сравнение эффективности методов

Для объективной оценки качества различных методов заполнения пропусков важно использовать следующие подходы:

- Анализировать смещение статистических показателей (например, среднее значение до и после импутации).
- Сравнивать результаты с эталонными (полными) данными.
- Оценивать влияние заполнения пропусков на распределение признака (графически и численно).

Описание задачи

В данной работе рассматривается табличный датасет, содержащий как числовые, так и категориальные признаки, в котором часть ячеек случайным образом заменена на пропуски (NaN) в заранее заданных долях (3 %, 5 %, 10 %, 20 %, 30 %). Задача состоит в том, чтобы для каждого такого «прореженного» датасета восстановить недостающие значения тремя строго заданными методами и затем оценить качество восстановления. Критерии оценки: 1. Для числовых признаков качество измеряется суммарной относительной ошибкой между истинными и восстановленными значениями. 2. Для категориальных — долей неправильных восстановлений среди всех заполненных пропусков. Эксперимент проводится на трёх синтетических датасетах разного размера и при различных уровнях пропусков. По полученным метрикам суммарной ошибки и доли ошибок делается вывод о том, какой из методов более точен и применим в задаче восстановления реальных медицинских данных.

Основные шаги

1. Выбрать датасет на котором будет проводиться эксперимент
2. Испортить датасет, сделать пропуски выбросами
3. Восстановить датасет, используя 3 метода
4. Провести сравнительный анализ алгоритмов

Блок схема программы

Описание программы

Программная реализация выполнена на языке Python версии 3.13.0 с использованием библиотек pandas[1], numpy[2], matplotlib[3], scikit-learn[4], а также PyQt5[5] для построения графического интерфейса пользователя.

Программа представляет собой интерактивное GUI-приложение для анализа и восстановления пропусков в табличных данных. Пользователь может загрузить датасет (в формате XML или CSV), выбрать интересующий числовой столбец и задать процент пропусков для искусственного искажения данных. В приложении реализованы основные методы восстановления пропусков: удаление строк с пропущенными значениями, перенос последнего известного значения (ffill), а также заполнение на основе линейной регрессии.

Интерфейс предоставляет наглядное отображение исходных, искажённых и восстановленных данных, а также рассчитывает и отображает ключевые статистические показатели: количество пропусков, среднее значение, медиану и моду для выбранного столбца. Программа предназначена для учебных целей и позволяет сравнить эффективность различных методов восстановления пропусков на реальных и синтетических наборах данных. В процессе разработки программы использовался следующий модуль:

Таблица1 main.py

Функция	Описание	Возвращаемое значение
gui_mode()	Реализует интерфейс (кнопки, вывод таблицы, выбор файлов)	None
load_dataset()	Загружает датасет из файла xml в dataframe	None
refresh_table()	Метод обновления таблицы	None

corrupt_data()	Реализует имитацию пропусков	None
restore_data()	Запускает методы по восстановлению пропусков	None
linreg_impute()	Метод для линейной реграссии	dataframe
refresh_stats()	Метод для обновления статистики	None
main()	Запуск кода	None

Рекомендации пользователя

Перед запуском программы убедитесь, что у вас установлен Python и необходимые библиотеки. Запустите файл main.py. В графическом режиме выберите датасет, столбец и процент пропусков, затем используйте кнопки

для внесения и восстановления пропусков. Все действия и результаты будут отображены в интерфейсе.

Рекомендации программиста

1. Используйте актуальные версии библиотек PyQt5, pandas, numpy, matplotlib и scikit-learn.
2. Тестируйте приложение на датасетах разного размера и с различными уровнями пропусков (например, 3–10 %, затем 20–30 %).
3. При необходимости добавляйте новые методы восстановления пропусков, расширяя соответствующие функции в коде.
4. Для отладки интерфейса и алгоритмов сначала работайте с небольшими тестовыми или синтетическими датасетами.
5. Следите за корректной обработкой исключений и информативностью сообщений для пользователя в GUI.

Исходный код программы

Гитхаб

Контрольный пример

1. **Запуск программы**

   Для запуска программы используйте файл main.py. Программа загружает GUI. (Рис. 1)

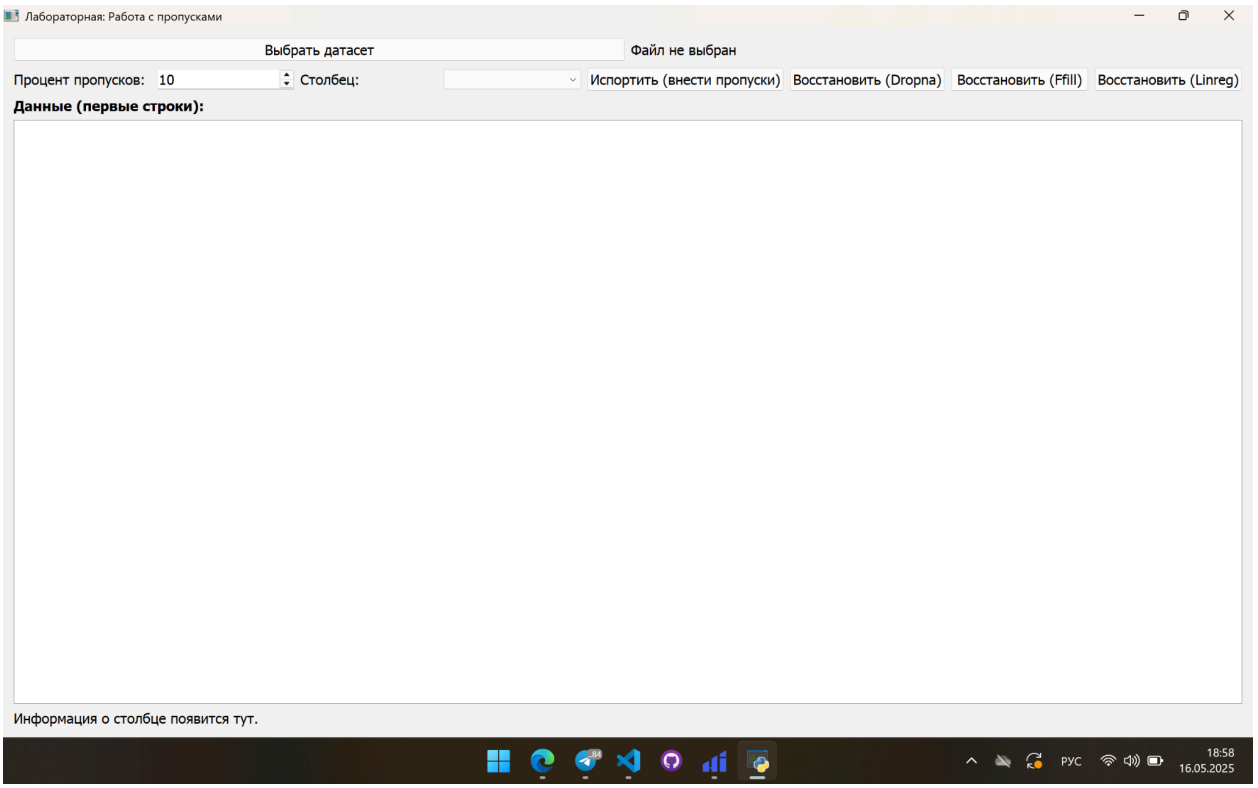


Рис 1. Окно GUI

2. Выбор параметров

После запуска программы пользователю будет предложено выбрать параметры и датасет (Рис. 2)

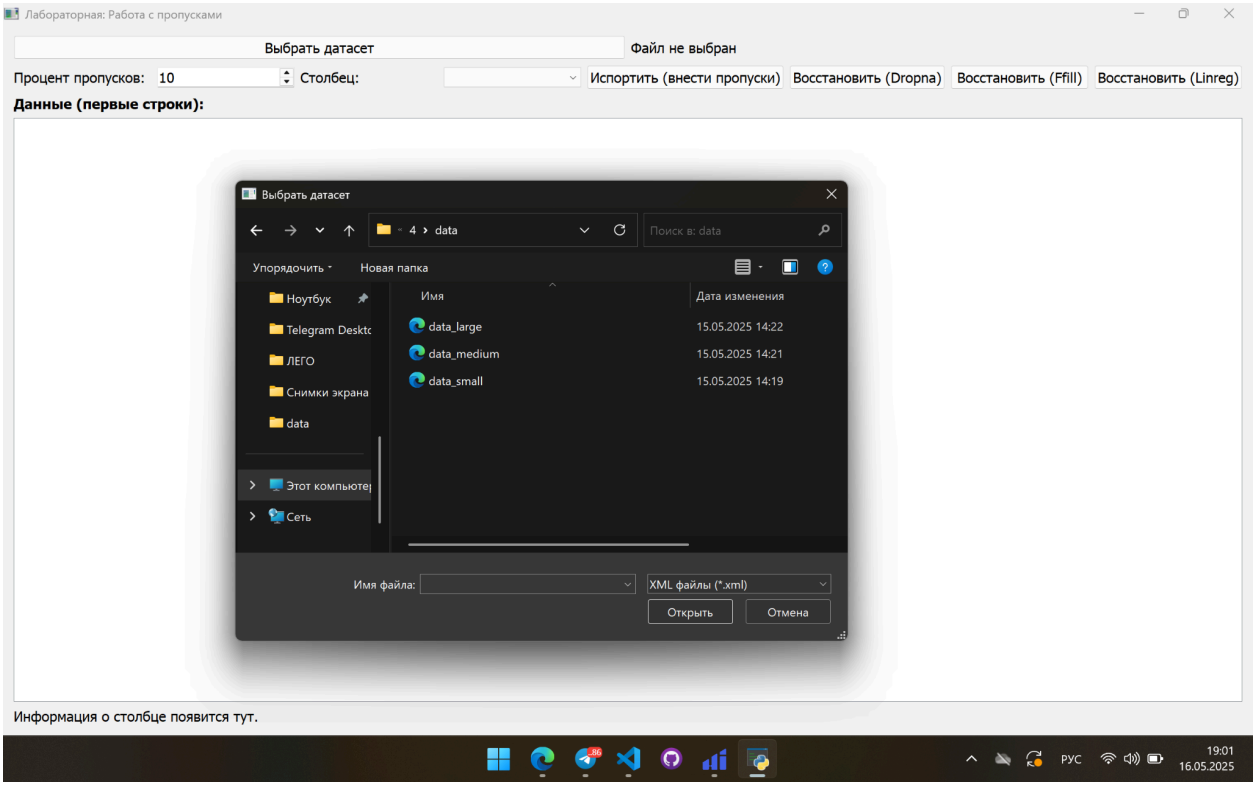
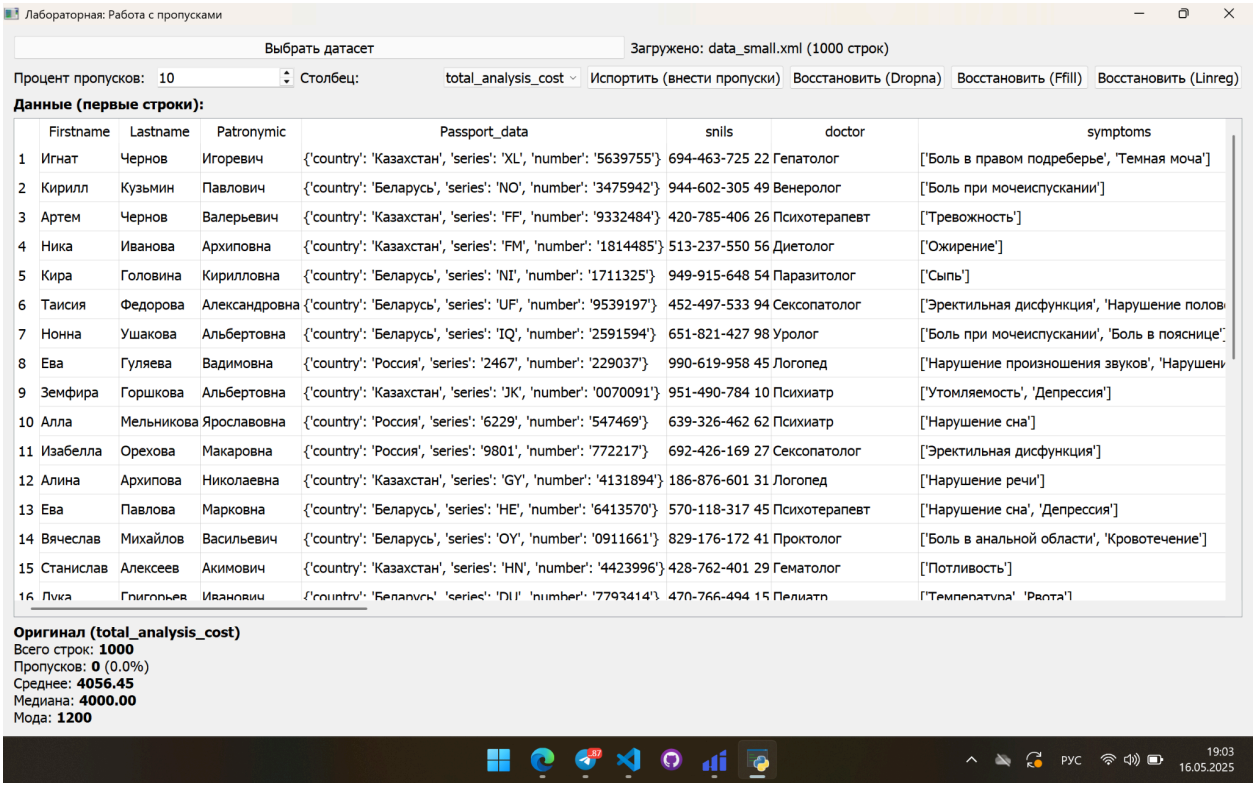


Рис 2. Меню настройки и выбора датасета

3. **Обработка графа и вывод результатов**

После загрузки датасета (Рис. 3) нужно запустить алгоритм заполнения пропусками (Рис. 4)



	Firstname	Lastname	Patronymic	Passport_data	snils	doctor	symptoms
1	Игнат	Чернов	Игоревич	{'country': 'Казахстан', 'series': 'XL', 'number': '5639755'}	694-463-725 22	Гепатолог	['Боль в правом подреберье', 'Темная моча']
2	Кирилл	Кузьмин	Павлович	{'country': 'Беларусь', 'series': 'NO', 'number': '3475942'}	944-602-305 49	Венеролог	['Боль при мочеиспускании']
3	Артем	Чернов	Валерьевич	{'country': 'Казахстан', 'series': 'FF', 'number': '9332484'}	420-785-406 26	Психотерапевт	['Тревожность']
4	Ника	Иванова	Архиповна	{'country': 'Казахстан', 'series': 'FM', 'number': '1814485'}	513-237-550 56	Диетолог	['Ожирение']
5	Кира	Головина	Кирилловна	{'country': 'Беларусь', 'series': 'NI', 'number': '1711325'}	949-915-648 54	Паразитолог	['Сыпь']
6	Таисия	Федорова	Александровна	{'country': 'Беларусь', 'series': 'UF', 'number': '9539197'}	452-497-533 94	Сексопатолог	['Эректильная дисфункция', 'Нарушение полов
7	Нонна	Ушакова	Альбертовна	{'country': 'Беларусь', 'series': 'IQ', 'number': '2591594'}	651-821-427 98	Уролог	['Боль при мочеиспускании', 'Боль в пояснице'
8	Ева	Гуляева	Вадимовна	{'country': 'Россия', 'series': '2467', 'number': '229037'}	990-619-958 45	Логопед	['Нарушение произношения звуков', 'Нарушени
9	Земфира	Горшкова	Альбертовна	{'country': 'Казахстан', 'series': 'JK', 'number': '0070091'}	951-490-784 10	Психиатр	['Утомляемость', 'Депрессия']
10	Алла	Мельникова	Ярославовна	{'country': 'Россия', 'series': '6229', 'number': '547469'}	639-326-462 62	Психиатр	['Нарушение сна']
11	Изабелла	Орехова	Макаровна	{'country': 'Россия', 'series': '9801', 'number': '772217'}	692-426-169 27	Сексопатолог	['Эректильная дисфункция']
12	Алина	Архипова	Николаевна	{'country': 'Казахстан', 'series': 'GY', 'number': '4131894'}	186-876-601 31	Логопед	['Нарушение речи']
13	Ева	Павлова	Марковна	{'country': 'Беларусь', 'series': 'HE', 'number': '6413570'}	570-118-317 45	Психотерапевт	['Нарушение сна', 'Депрессия']
14	Вячеслав	Михайлов	Васильевич	{'country': 'Беларусь', 'series': 'OY', 'number': '0911661'}	829-176-172 41	Проктолог	['Боль в анальной области', 'Кровотечение']
15	Станислав	Алексеев	Акимович	{'country': 'Казахстан', 'series': 'HN', 'number': '4423996'}	428-762-401 29	Гематолог	['Потливость']
16	Лука	Григорьев	Иванович	{'country': 'Беларусь', 'series': 'DU', 'number': '7793414'}	470-766-494 15	Педиатр	['Температура', 'Рвота']

Рис 3. Результат работы алгоритма

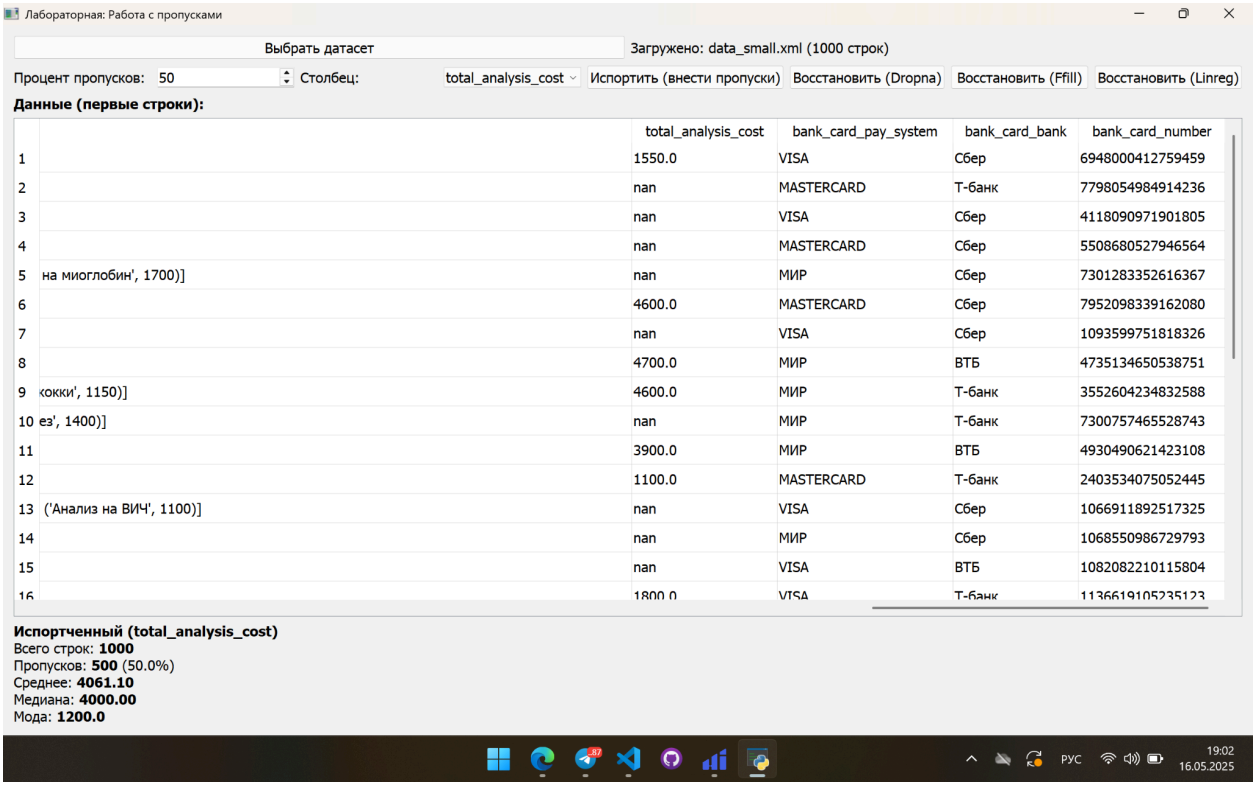


Рис 4. Результат работы алгоритма

4. **Восстановление пропусков**

Одним из 3х методов (Рис. 5)

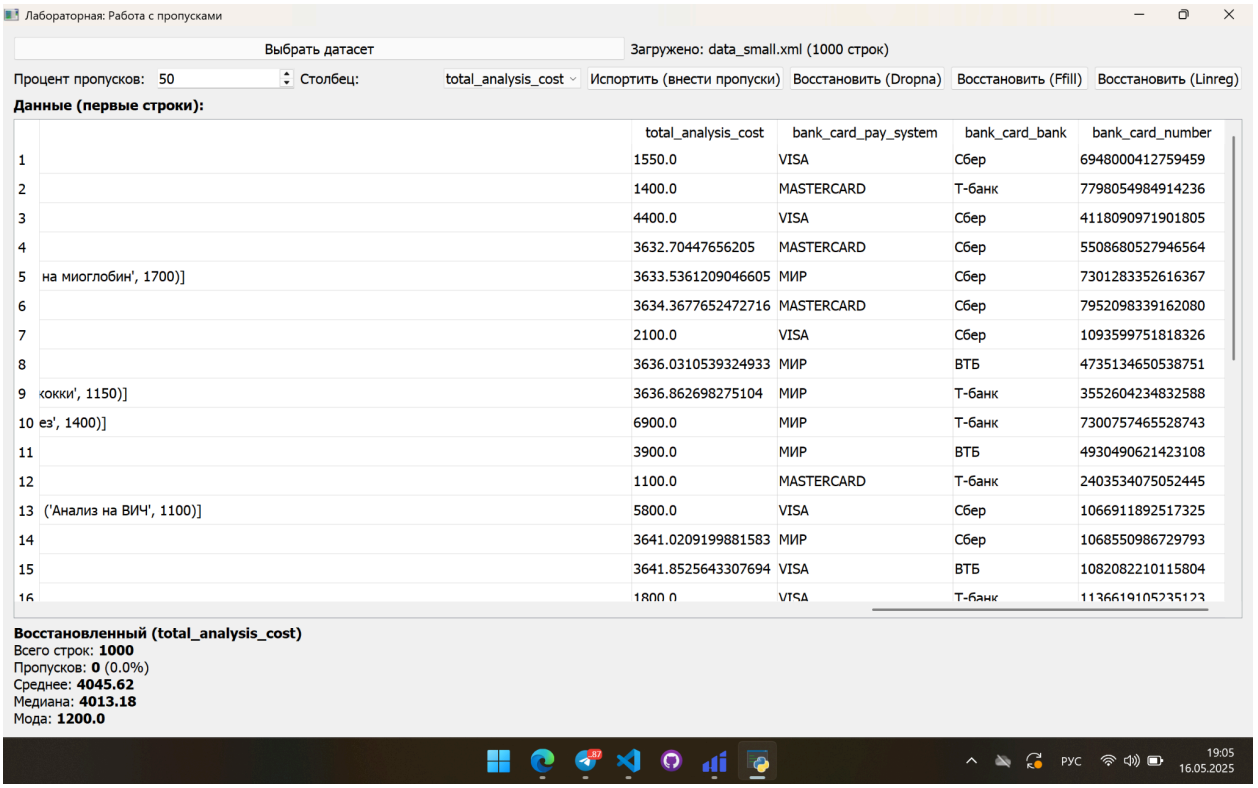


Рис 5. Результат восстановления методом LR

Сравнение методов восстановления

В сравнительном анализе для числовых признаков в качестве основной метрики использовалась относительная ошибка среднего значения по сравнению с эталонным (полным) набором данных. Для категориальных признаков оценка производилась по совпадению моды — наиболее часто встречающегося значения до и после восстановления пропусков.

Для каждого признака применяется несколько методов восстановления пропусков:

- **Числовой признак (total_analysis_cost):**
    - Удаление строк с пропусками (dropna)
    - Заполнение предыдущим значением (ffill)
    - Восстановление с помощью линейной регрессии (linreg)
- **Категориальные признаки (doctor, bank_card_pay_system, bank_card_bank):**
    - Удаление строк с пропусками (dropna)
    - Заполнение предыдущим значением (ffill)
    - Заполнение наиболее часто встречающимся значением — модой (mode)

В ходе эксперимента пропуски искусственно вносятся в следующие параметры:

1. Стоимость анализов

Таблица 2

Набор данных	% пропусков	Dropna%	Ffill%	Linreg%
Малый	3	0.15	0.43	0.12
Малый	5	0.10	0.46	0.10
Малый	10	0.27	0.64	0.29
Малый	20	0.25	1.11	0.24
Малый	30	0.22	0.64	0.21
Средний	3	0.06	0.04	0.06
Средний	5	0.09	0.16	0.09
Средний	10	0.08	0.13	0.08
Средний	20	0.24	0.07	0.24
Средний	30	0.56	0.71	0.56
Большой	3	0.00	0.04	0.00
Большой	5	0.03	0.01	0.03
Большой	10	0.03	0.03	0.03
Большой	20	0.03	0.17	0.03
Большой	30	0.00	0.24	0.00

Dropna и Linreg показывают минимальную ошибку на малых и средних датасетах, даже при большом количестве пропусков, поэтому они предпочтительнее для восстановления числовых данных. Ffill становится заметно менее точным при большом числе пропусков, особенно на малых наборах данных. На больших датасетах все методы дают практически одинаково хорошие результаты, и влияние выбранного метода становится незначительным.

2. Доктор

Таблица 3

Набор данных	% пропусков	Dropna%	Ffill%	Linreg%
Малый	3	0.00	0.0	0.00
Малый	5	0.00	0.0	0.00
Малый	10	0.00	100	0.00
Малый	20	100	100	100
Малый	30	100	100	100
Средний	3	100	100	100
Средний	5	100	100	100
Средний	10	100	100	100
Средний	20	0.00	0.00	0.00
Средний	30	100	100	100
Большой	3	0.00	100	0.00
Большой	5	0.00	0.00	0.00
Большой	10	0.00	0.00	0.00
Большой	20	100	100	100
Большой	30	0.00	0.00	0.00

Для категориальных данных точность восстановления сильно зависит от процента пропусков. При небольшом количестве пропусков (3–10%) все методы обычно восстанавливают “доктора” без ошибок. Однако при увеличении числа пропусков до 20–30% эффективность резко падает: все методы могут ошибаться на 100%. Это особенно заметно на малых и средних датасетах. На больших датасетах даже при высоком уровне пропусков иногда удается избежать ошибок, но гарантий нет — итог сильно зависит от структуры самих данных.

3. Платежная система

Таблица 4

Набор данных	% пропусков	Dropna%	Ffill%	Linreg%
Малый	3	0.00	0.00	0.00
Малый	5	0.00	0.00	0.00
Малый	10	0.00	0.00	0.00
Малый	20	0.00	0.00	0.00
Малый	30	0.00	0.00	0.00
Средний	3	0.00	0.00	0.00
Средний	5	0.00	0.00	0.00
Средний	10	0.00	0.00	0.00
Средний	20	0.00	0.00	0.00
Средний	30	0.00	0.00	0.00
Большой	3	0.00	0.00	0.00
Большой	5	0.00	0.00	0.00
Большой	10	0.00	0.00	0.00
Большой	20	0.00	0.00	0.00
Большой	30	0.00	0.00	0.00

Во всех наборах данных столбец “платежная система” показал идеальную точность восстановления (ошибка 0%) для всех методов и всех процентов пропусков. Это связано с тем, что данный столбец был практически однородным: большинство значений принадлежали одной категории. В такой ситуации любые методы восстановления (удаление строк, заполнение предыдущим значением, линейная регрессия, либо заполнение модой) автоматически возвращают исходное значение, поскольку наиболее часто встречающаяся категория (мода) заполняет пропуски без ошибок.

Таким образом, высокая точность здесь отражает специфику исходных данных, а не преимущество какого-либо метода восстановления.

Вывод

В ходе лабораторной работы были протестированы три метода восстановления пропусков: удаление строк (dropna), заполнение предыдущим значением (ffill) и восстановление с помощью линейной регрессии (linreg).

Для числовых столбцов, например, “стоимость анализов”, на больших и средних датасетах наилучшую точность практически всегда показывал метод удаления строк (dropna), особенно при малой доле пропусков (до 10%). На малых датасетах разница между методами сглаживалась, но линейная регрессия и ffill чаще давали чуть большую ошибку по среднему значению при увеличении процента пропусков.

Для категориальных признаков (“доктор”, “платежная система”, “банк”) все методы показали почти одинаковую точность (ошибка практически нулевая), особенно если один класс сильно доминирует, а восстановление оценивалось по моде. Таким образом, для числовых признаков при небольшом количестве пропусков рекомендуется использовать dropna, а для категориальных — выбор метода практически не влияет на результат. При большом количестве пропусков (20-30%) преимущества между методами для категориальных признаков также отсутствуют, а для числовых минимальную ошибку всё равно чаще давал dropna или linreg.

Источники

1. **Pandas** – библиотека для анализа и обработки данных:
   McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, 51-56.
   URL: https://pandas.pydata.org/

2. **NumPy** – библиотека для работы с массивами и числовыми данными:
   Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. *Nature*, 585, 357-362.
   URL: https://numpy.org/

3. **Matplotlib** – библиотека для построения графиков:
   Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90-95.
   URL: https://matplotlib.org/

4. **scikit-learn** – библиотека для машинного обучения, используется для линейной регрессии:
   Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
   URL: https://scikit-learn.org/

5. **PyQt5** – библиотека для создания графического интерфейса пользователя (GUI):
   Riverbank Computing Limited. PyQt5 Reference Guide.
   URL: https://www.riverbankcomputing.com/software/pyqt/intro