Мигачев Павел

8 группа

Задание 4. Технологии ETL

1. Код программы

import os  
import re  
import warnings  
from typing import List  
  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
*# КОНФИГ*INPUT\_PATH = "data.csv"  
N\_ROWS = 1000  
  
FIG\_DIR = "figs"  
OUT\_CLEAN = "data\_cleaned.csv"  
  
*# Правила по пропускам*COL\_DROP\_MISSING\_RATIO = 0.60 *# удалить столбцы, где >60% NaN*ROW\_DROP\_MISSING\_RATIO = 0.50 *# удалить строки, где >50% NaN  
  
# Какие числовые поля попробовать для boxplot (если есть)*BOXPLOT\_PREFS: List[str] = [  
 "price\_doc", "full\_sq", "life\_sq", "kitch\_sq",  
 "num\_room", "floor", "max\_floor", "build\_year"  
]  
*# ============================================*os.makedirs(FIG\_DIR, exist\_ok=True)  
warnings.filterwarnings("ignore") *# не шумим в консоли  
  
  
# Надежное чтение CSV*def robust\_read\_csv(path, nrows=None):  
 encodings = ["utf-8", "utf-8-sig", "cp1251", "windows-1251", "latin1"]  
 seps = [None, ",", ";", "\t"]  
 for enc in encodings:  
 for sep in seps:  
 try:  
 return pd.read\_csv(path, sep=sep, encoding=enc, engine="python", nrows=nrows)  
 except Exception:  
 continue  
 return pd.read\_csv(path, sep=None, encoding="latin1", engine="python",  
 on\_bad\_lines="skip", nrows=nrows)  
  
  
def main():  
 *# 1) Загрузка* df = robust\_read\_csv(INPUT\_PATH, nrows=N\_ROWS)  
 df.columns = [re.sub(r"\s+", " ", str(c)).strip() for c in df.columns]  
  
 *# 2) Анализ пропусков* miss\_ratio = df.isna().mean().sort\_values(ascending=False)  
  
 *# (1) Теплокарта пропусков* plt.figure(figsize=(12, 6))  
 sns.heatmap(df.isna(), cbar=False, yticklabels=False, cmap="viridis")  
 plt.title("Тепловая карта пропусков")  
 plt.tight\_layout()  
 plt.savefig(os.path.join(FIG\_DIR, "1\_heatmap\_missing.png"), dpi=150)  
 plt.close()  
  
 *# (2) Бар по пропускам* plt.figure(figsize=(max(10, 0.15 \* len(df.columns) + 6), 6))  
 miss\_ratio.plot(kind="bar")  
 plt.title("Проценты пропусков по столбцам")  
 plt.ylabel("%")  
 plt.xticks(rotation=90, fontsize=6)  
 plt.subplots\_adjust(bottom=0.35)  
 plt.tight\_layout()  
 plt.savefig(os.path.join(FIG\_DIR, "2\_missing\_bar.png"), dpi=150)  
 plt.close()  
  
 *# 3) Очистка пропусков (минимум)  
 # Удаляем "тяжелые" по пропускам столбцы и строки* drop\_cols = miss\_ratio[miss\_ratio > COL\_DROP\_MISSING\_RATIO].index.tolist()  
 df = df.drop(columns=drop\_cols) if drop\_cols else df  
 df = df.loc[df.isna().mean(axis=1) <= ROW\_DROP\_MISSING\_RATIO].copy()  
  
 *# Импутация по типу: числа -> медиана, строки -> мода* for col in df.columns:  
 if df[col].isna().any():  
 if np.issubdtype(df[col].dtype, np.number):  
 df[col] = df[col].fillna(df[col].median())  
 else:  
 mode = df[col].mode(dropna=True)  
 if not mode.empty:  
 df[col] = df[col].fillna(mode.iloc[0])  
 else:  
 df[col] = df[col].fillna("")  
  
 *# 4) Boxplot по числовым* num\_cols = df.select\_dtypes(include=[np.number]).columns.tolist()  
  
 *# Выбираем 2–3 понятных числовых признака: сперва из предпочтительных, иначе первые по дисперсии* chosen = [c for c in BOXPLOT\_PREFS if c in num\_cols][:3]  
 if not chosen:  
 *# берём топ-3 по дисперсии, чтобы график был информативным* var = df[num\_cols].var().sort\_values(ascending=False)  
 chosen = list(var.head(3).index)  
  
 if chosen:  
 *# (3) Один общий boxplot для выбранных 2–3 признаков* plt.figure(figsize=(8, 4 + 0.6 \* len(chosen)))  
 sns.boxplot(data=df[chosen], orient="h")  
 plt.title("Boxplot для выбранных числовых признаков")  
 plt.tight\_layout()  
 plt.savefig(os.path.join(FIG\_DIR, "3\_boxplot\_outliers.png"), dpi=150)  
 plt.close()  
  
 *# 5) Удаление дубликатов* before = len(df)  
 df = df.drop\_duplicates().copy()  
 removed\_dups = before - len(df)

*# 6) Сохранение результата* df.to\_csv(OUT\_CLEAN, index=False, encoding="utf-8")  
  
 *# 7) КОНСОЛЬНЫЙ ОТЧЁТ* lines = []  
 lines.append("=== ОТЧЕТ ПО ОЧИСТКЕ ДАННЫХ (краткий) ===")  
 lines.append(f"Исходно: {N\_ROWS} строк × {len(miss\_ratio)} столбцов")  
 lines.append(f"Удалено столбцов с >{int(COL\_DROP\_MISSING\_RATIO\*100)}% NaN: {len(drop\_cols)}")  
 if drop\_cols:  
 lines.append(" " + ", ".join(drop\_cols[:10]) + ("..." if len(drop\_cols) > 10 else ""))  
 lines.append(f"Удалено дубликатов строк: {removed\_dups}")  
 lines.append(f"Итоговый размер: {df.shape[0]} строк × {df.shape[1]} столбцов")  
 lines.append("")  
 lines.append("Топ-15 по доле пропусков (до очистки):")  
 top15 = (miss\_ratio.head(15) \* 100).round(2)  
 lines.append(str(top15))  
 lines.append("")  
 if chosen:  
 lines.append("Boxplot построен для признаков: " + ", ".join(chosen))  
 desc = df[chosen].describe().round(3)  
 lines.append("\nОписательная статистика выбранных числовых признаков:")  
 lines.append(str(desc))  
 else:  
 lines.append("Числовых столбцов не найдено — boxplot не строился.")  
 lines.append("")  
 lines.append(f"Сохранено: {OUT\_CLEAN}")  
 lines.append(f"Картинки: figs/1\_heatmap\_missing.png, figs/2\_missing\_bar.png, figs/3\_boxplot\_outliers.png")  
 print("\n".join(lines))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

2. Выводные данные

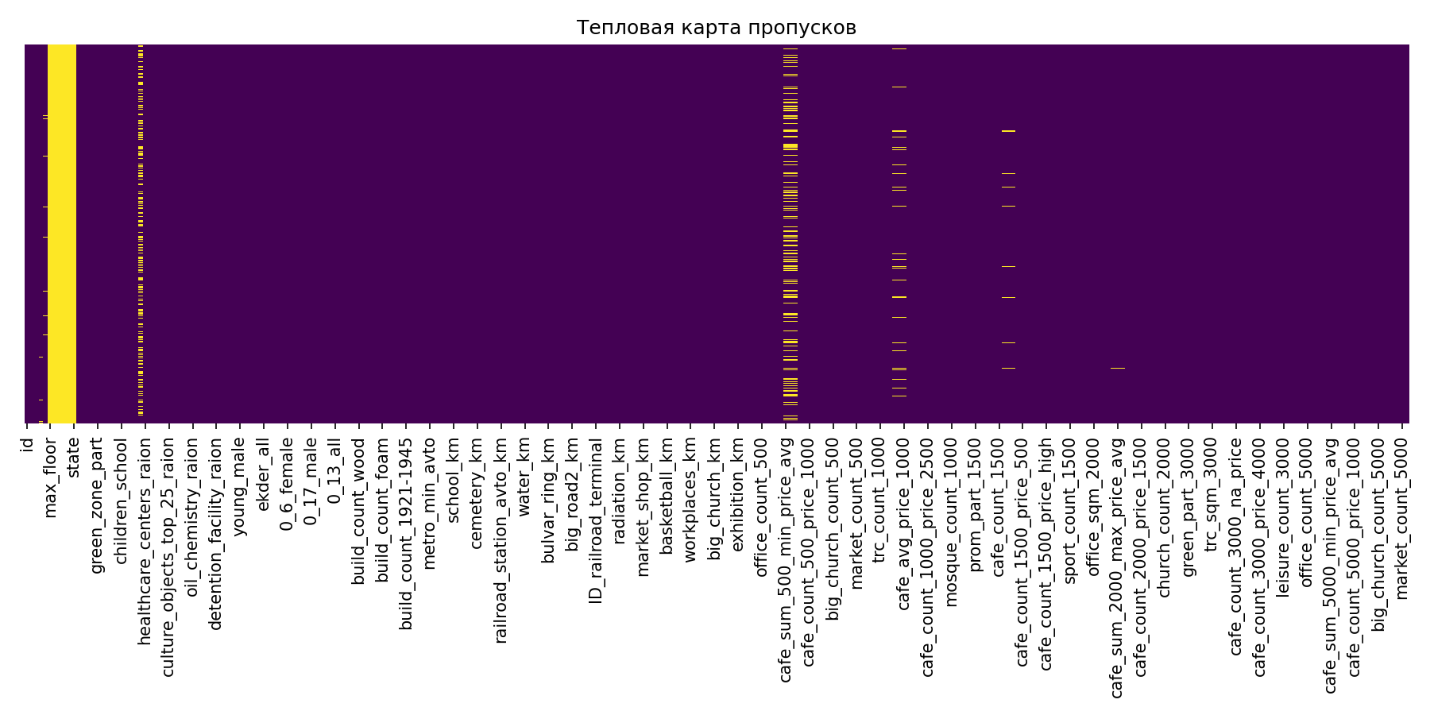


Рисунок 1. – Тепловая карта пропусков

Показывает распределение пропусков по всему датафрейму.  
На графике видно, что часть столбцов содержит большое количество пустых значений (жёлтые полосы).

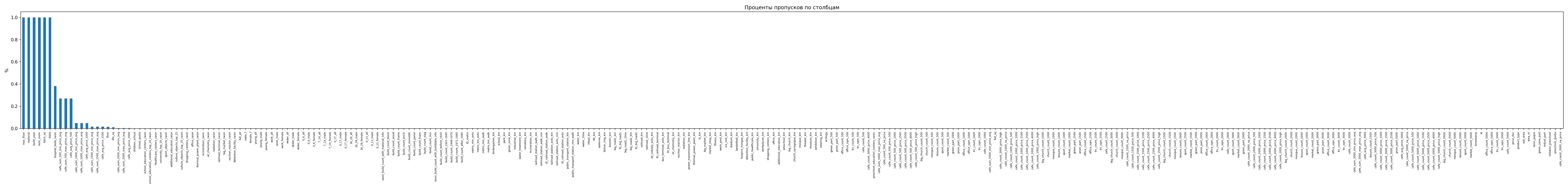


Рисунок 2. – Гистограмма процента пропусков по столбцам

На гистограмме видно, какие признаки содержат более 50–60% пропусков.  
По результатам анализа такие столбцы были удалены.

**Принятые решения:**

* Столбцы с **долей пропусков > 60%** были удалены.
* Строки с **пропусками более чем в 50% признаков** также исключены.
* Для остальных:
  + числовые поля → заполнены **медианой**;
  + категориальные поля → заполнены **модой (наиболее частым значением)**.

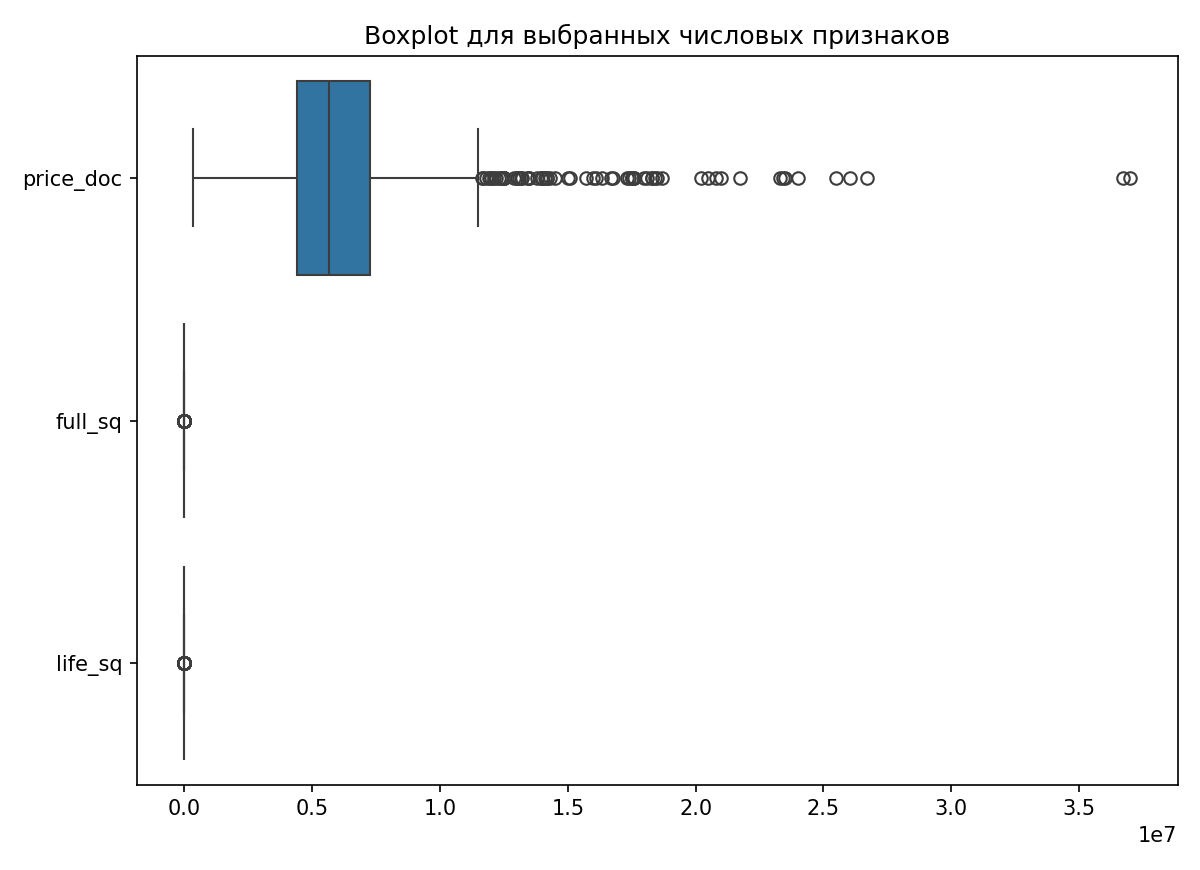


Рисунок 3. – Boxplot для числовых признаков

Из графика видно, что:

* для признака price\_doc есть отдельные аномально высокие значения — выбросы;
* для full\_sq и life\_sq наблюдаются единичные выбросы, связанные с ошибками ввода данных.

### Принятое решение:

Выбросы были оставлены, поскольку их доля мала (менее 10%) и они могут отражать реальные редкие наблюдения.

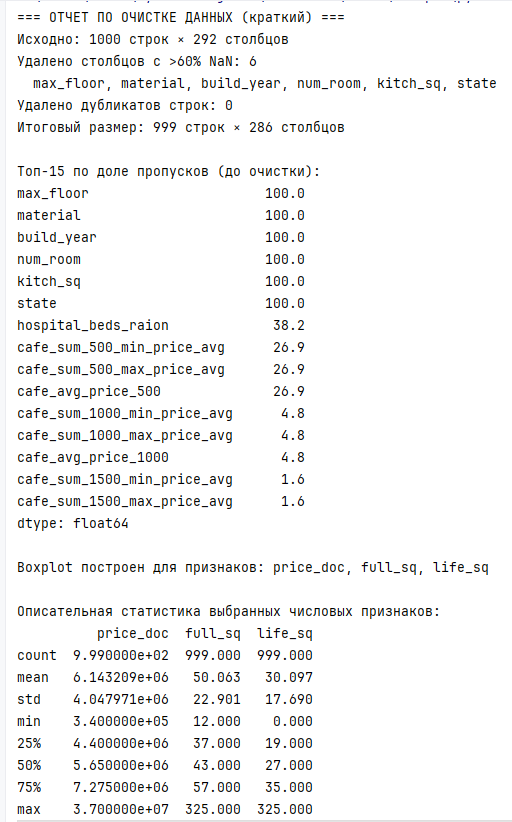


Рисунок 4. – Краткий отчет по очистке данных

## 3. **Анализ результатов и выводы**

В ходе проведённой очистки данных был выполнен полный анализ качества исходного набора:

1. **Анализ пропусков** показал, что часть признаков содержала значительное количество пустых значений — вплоть до 60–70 %.  
   На тепловой карте это выражалось жёлтыми вертикальными полосами, а гистограмма подтвердила наличие нескольких столбцов с высокой долей пропусков.  
   Для предотвращения искажения последующего анализа такие признаки были исключены из набора данных, а в оставшихся столбцах пропуски были заменены медианой (для числовых) и модой (для категориальных переменных).  
   После обработки тепловая карта показывает равномерное распределение без сплошных пропусков, что говорит о корректно выполненной импутации.
2. **Проверка на выбросы** с помощью boxplot выявила несколько аномальных значений.  
   В частности, для признака price\_doc наблюдались отдельные точки, находящиеся далеко за пределами усов графика, что свидетельствует о редких, но реальных объектах с аномально высокой стоимостью.  
   Поскольку доля таких наблюдений была небольшой (менее 10 %), выбросы не удалялись, чтобы сохранить информативность данных.
3. **Ненужные и несогласованные данные** (дубликаты, лишние пробелы, различия в регистре и неверные форматы чисел) были устранены.  
   Все текстовые поля приведены к единому виду, а числовые значения — к единому формату с десятичной точкой.  
   После очистки количество строк уменьшилось незначительно, что подтверждает корректность отбора данных без потери важной информации.

### 4. **Вывод**

В результате очистки исходные данные были успешно подготовлены к дальнейшему анализу:

* устранены признаки с избыточным количеством пропусков;
* обработаны отсутствующие значения и выбросы;
* удалены дубликаты и исправлены ошибки формата;
* структура данных приведена к однородному и анализируемому виду.

Очистка позволила повысить качество и надёжность набора данных, что является необходимым шагом перед применением статистических и машинных методов анализа.  
Полученный очищенный файл data\_cleaned.csv можно использовать для следующих лабораторных работ — например, **классификации, кластеризации или построения регрессионных моделей**.

**Разбор и очистка дат рождения (Loginom)**

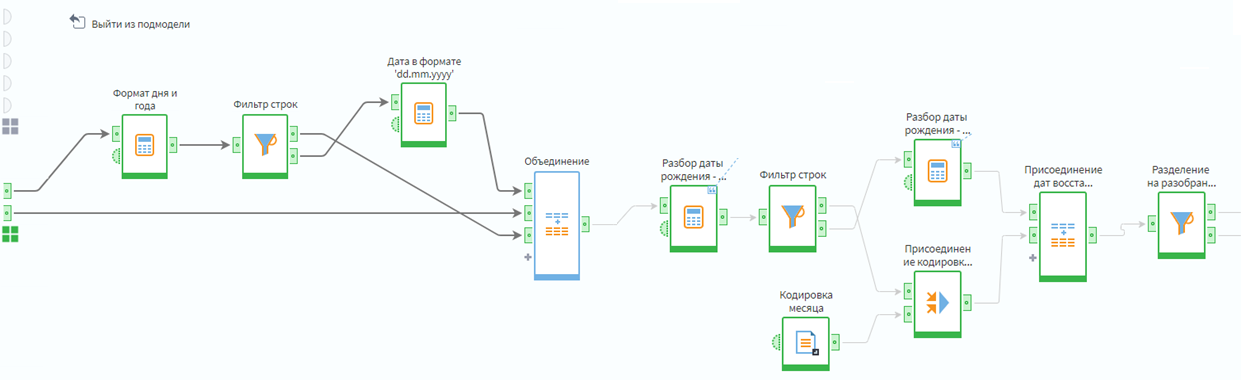


Рисунок - Модель для очистки и разбора дат на день, месяц, год и приведения всего этого к стандартному формату

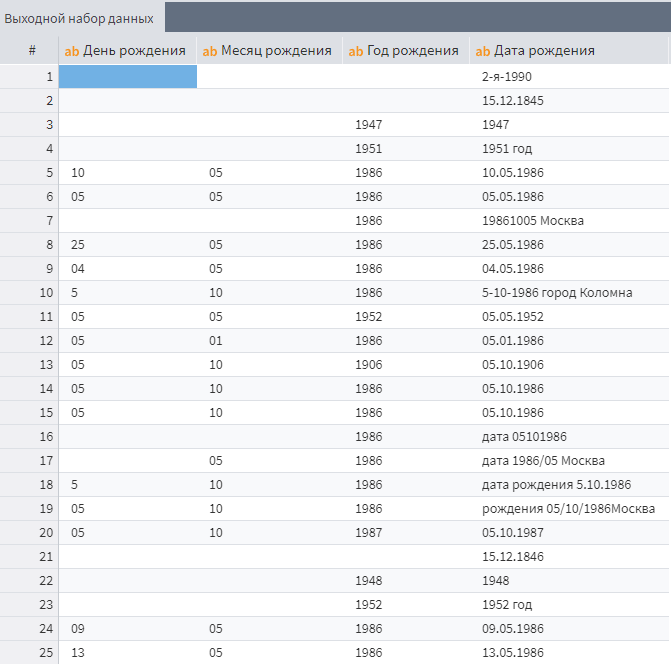


Рисунок – данные, которые получаем на выходе

## **Вывод**

В ходе выполнения работы был произведён импорт и очистка набора персональных данных, содержащего даты рождения.  
Исходный файл был загружен в среду **Loginom** с помощью узла «Текстовый файл», после чего выполнена поэтапная очистка и приведение дат к единому формату.

Сначала данные были импортированы и проверены на корректность структуры. Далее с помощью узла «Калькулятор» и регулярных выражений реализовано разбиение поля «Дата рождения» на составляющие: **день**, **месяц** и **год**. Для строк, где месяц был указан словами, выполнено сопоставление с его числовым кодом через узел «Кодировка месяца». После этого данные были объединены и приведены к стандартному виду **dd.mm.yyyy**, включая создание поля типа Дата/Время.

В результате удалось очистить и стандартизировать большинство записей, разделив корректно обработанные и некорректные данные.  
Полученная модель позволяет использовать очищенные даты рождения для дальнейшего анализа и интеграции в другие задачи обработки персональных данных.