

# 1 Введение

**Цель работы:** Закрепить навыки работы с библиотекой Pandas, провести разведочный анализ данных (EDA), визуализировать ключевые зависимости и освоить взаимодействие с базой данных SQLite через Python.

**Описание набора данных:** В работе используется классический датасет *German Credit Data* (Statlog). Он содержит 1000 записей о клиентах банка. Каждая запись описывается 20 атрибутами (как числовыми, так и категориальными) и целевой переменной, указывающей на кредитный риск.

**Целевая переменная (Risk):**

- 1 (Good) — надежный заемщик.
- 2 (Bad) — ненадежный заемщик (дефолт).

Также учитывается матрица стоимости (Cost Matrix), согласно которой ошибка классификации "плохого" клиента как "хорошего" стоит в 5 раз дороже, чем наоборот.

# 2 Загрузка и подготовка данных

Данные были загружены из репозитория UCI. Так как исходный файл не содержал заголовков, они были назначены вручную согласно документации.

Одной из ключевых проблем датасета является кодировка категориальных признаков (например, A11, A30). Для качественного анализа было произведено декодирование с использованием словарей.

```
1 #  
2 map_checking = {  
3     'A11': '< 0 DM',  
4     'A12': '0 <= ... < 200 DM',  
5     'A13': '>= 200 DM',  
6     'A14': 'No checking account'  
7 }  
8 #  
9 df_readable['checking_status'] = df_readable['checking_status'].map(  
    map_checking)
```

Листинг 1: Фрагмент кода декодирования данных

**Результат подготовки:**

- Размерность данных: (1000 строк, 21 столбец).
- Пропущенные значения (NaN): отсутствуют.
- Добавлен столбец `risk_status` ('Good'/'Bad') для удобства.

# 3 Статистический анализ

Был проведен анализ распределений и баланса классов.

**Основные показатели:**

- Средняя сумма кредита: 3271 DM.
- Средний возраст заемщика: 35.5 лет.

- **Баланс классов:** 700 Good (70%) / 300 Bad (30%). Выборка несбалансированная.

**Анализ финансового риска:** Суммарный объем "плохих" кредитов составляет более 1.18 млн DM. Средняя сумма дефолтного кредита ( 3938 DM) выше средней суммы по всей выборке. Это подтверждает гипотезу, что чем выше сумма, тем выше риск.

## 4 Визуализация данных

### 4.1 Влияние статуса счета на риск

Самым сильным предиктором надежности оказался статус текущего счета (`checking_status`).

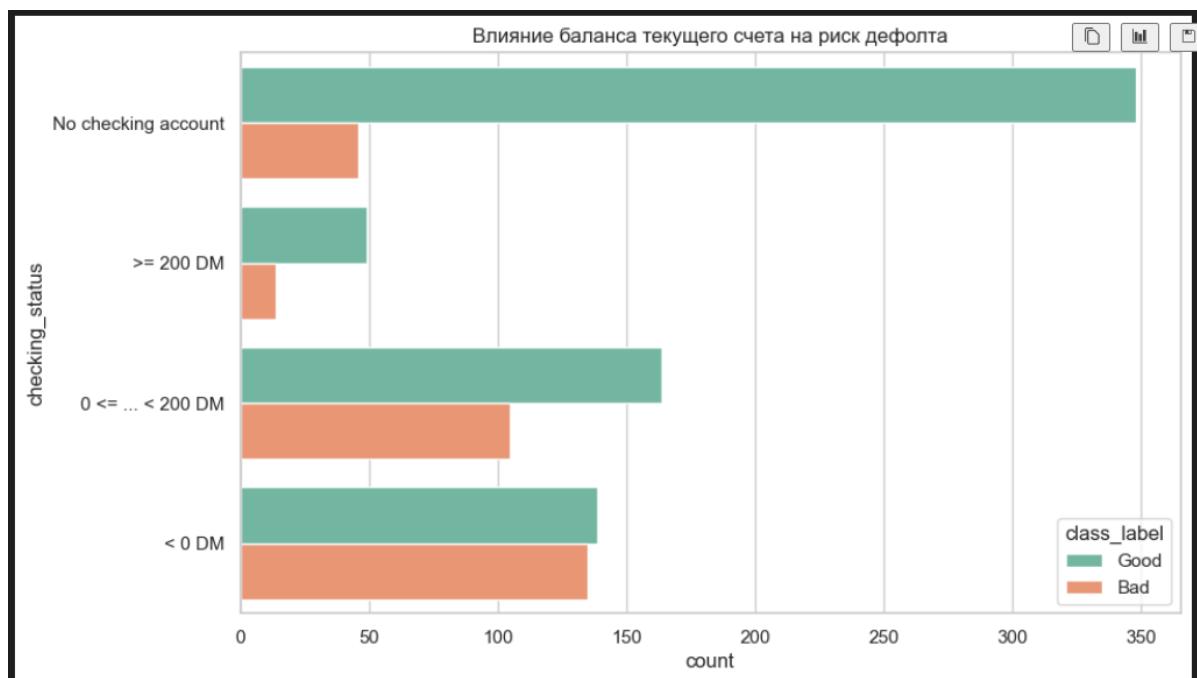


Рис. 1: Зависимость риска от баланса на счете

**Интерпретация:** Клиенты категории *No checking account* (нет текущего счета) демонстрируют наивысшую надежность. Напротив, клиенты с отрицательным балансом (*< 0 DM*) имеют вероятность дефолта около 50%, что делает эту категорию высокорисковой.

### 4.2 Распределение сумм кредита

Гистограмма показывает логнормальное распределение сумм кредитов.

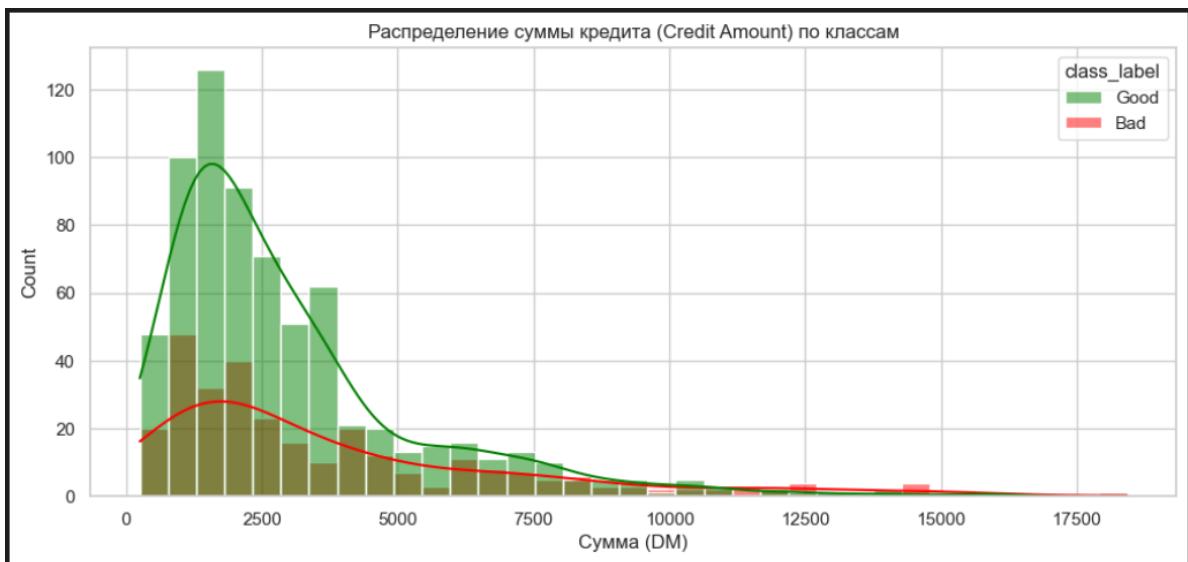


Рис. 2: Распределение суммы кредита (Good vs Bad)

**Интерпретация:** "Хорошие" заемщики сконцентрированы в области малых сумм (до 2500 DM). "Тяжелый хвост" распределения справа (крупные суммы) содержит значительную долю "плохих" кредитов.

### 4.3 Корреляционный анализ

Для построения матрицы корреляций категориальные признаки были закодированы методом LabelEncoder.

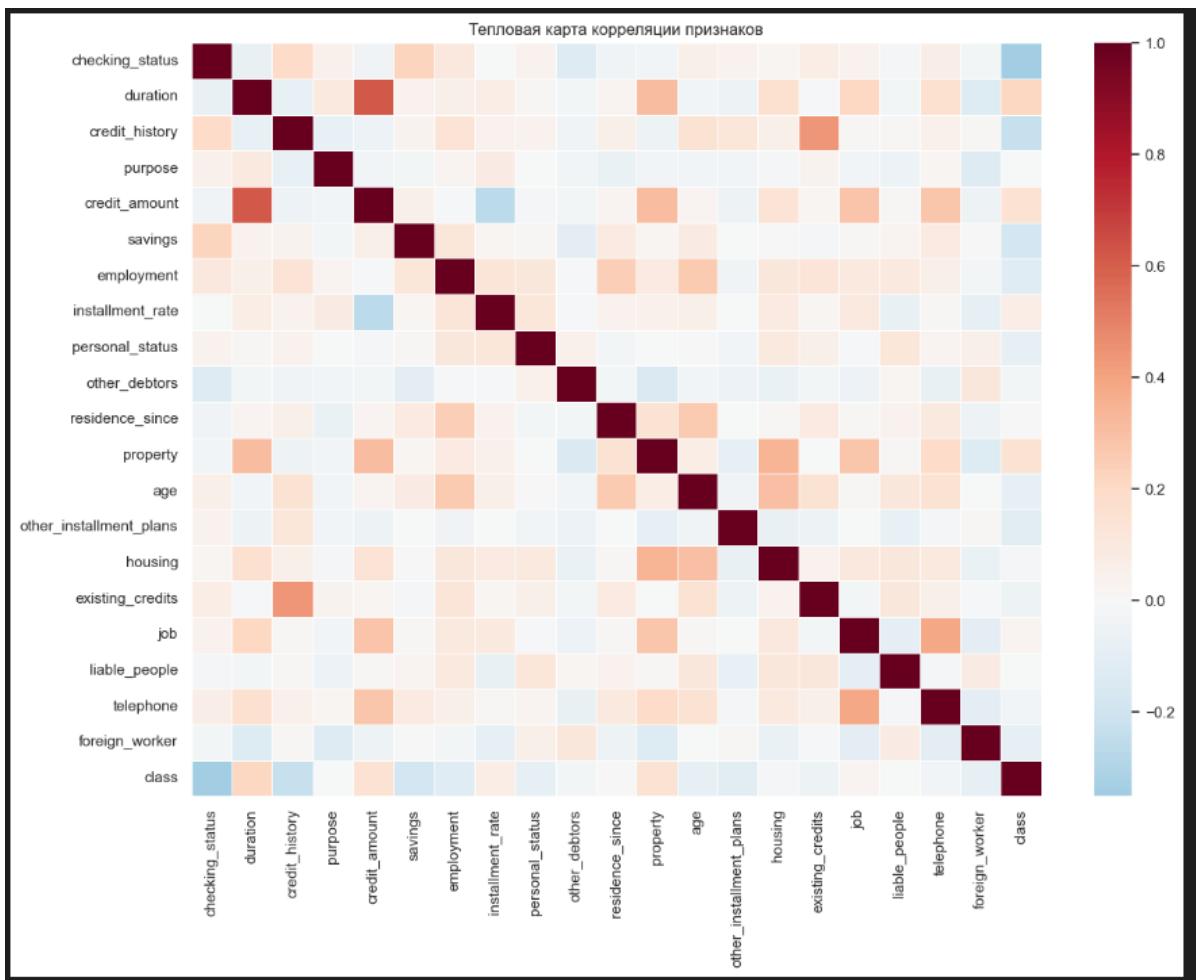


Рис. 3: Тепловая карта корреляций

**Интерпретация:** Наибольшую корреляцию с целевой переменной `class` показали признаки `checking_status` (0.35) и `credit_history` (0.23). Также выявлена логичная сильная связь между суммой кредита и его длительностью.

## 5 Работа с базой данных SQL

Обработанные данные были сохранены в локальную базу данных SQLite (`german_credit.db`).

### 5.1 Статистика по целям кредитования

Был выполнен SQL-запрос для агрегации данных по целям. Рассчитана средняя сумма и количество дефолтов.

```

1 SELECT
2     purpose, COUNT(*) as total,
3     ROUND(AVG(credit_amount), 0) as avg_amount,
4     SUM(CASE WHEN risk_status = 'Bad' THEN 1 ELSE 0 END) as bad_count
5 FROM credits
6 GROUP BY purpose
7 ORDER BY avg_amount DESC;
```

Листинг 2: SQL запрос агрегации

Результаты анализа (из таблицы `credits`):

Цель (Purpose)	Кол-во	Средняя сумма	Bad Count	% Bad
Others	12	8209	5	41.6%
Car (used)	103	5370	17	16.5%
Business	97	4158	34	35.0%
Education	50	3180	22	44.0%
Furniture/Equip	181	3067	58	32.0%
Car (new)	234	3063	89	38.0%
Radio/TV	280	2488	62	22.1%

Таблица 1: Агрегированные данные по целям

#### Вывод по таблице:

- Самая рискованная цель — **Education** (Образование). Доля невозвратов достигает 44%, несмотря на относительно небольшую среднюю сумму.
- Кредиты на подержанные автомобили (**Car used**) имеют высокую среднюю сумму (5370 DM), но низкий процент дефолтов (16.5%), что делает их выгодными для банка.
- Самая массовая категория — **Radio/TV** — является одной из самых надежных.

## 5.2 Поиск высокорисковых клиентов

Был выполнен запрос на поиск молодых заемщиков (до 35 лет) со статусом **Bad** и проблемами с текущим счетом.

```

1 SELECT age, job, credit_amount, checking_status, purpose
2 FROM credits
3 WHERE risk_status = 'Bad' AND age < 35
4   AND (checking_status LIKE '%< 0%', OR savings LIKE '%< 100%')
5 ORDER BY credit_amount DESC LIMIT 5;

```

В выборку попал клиент в возрасте 32 лет с суммой долга **18 424 DM** на "Другие цели" и клиент 23 лет с долгом **15 672 DM** на бизнес. Это демонстрирует необходимость более строгой проверки платежеспособности молодых клиентов, запрашивающих крупные суммы.

## 6 Заключение

В ходе выполнения лабораторной работы был проведен полный цикл анализа данных набора "German Credit Data".

- Подготовка данных:** Успешно загружен "сырой" датасет, проведено декодирование атрибутов, данные проверены на целостность.
- Анализ факторов риска:** Установлено, что наиболее значимым признаком надежности клиента является состояние его текущего счета (**Attribute 1**). Клиенты без счета надежнее клиентов с отрицательным балансом.
- Визуализация:** Построены графики, подтверждающие, что проблемные кредиты часто имеют более высокие суммы и сроки погашения.
- SQL-аналитика:** База данных позволила выявить скрытые инсайты. В частности, обнаружена аномально высокая рискованность образовательных кредитов (44% брака).

Работа демонстрирует, как методы Python и SQL могут применяться для построения систем банковского scoring и оценки рисков.