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.

```
#
map_checking = {
    'A11': '< 0 DM',
    'A12': '0 <= ... < 200 DM',
    'A13': '>= 200 DM',
    'A14': 'No checking account'
}
#
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`).

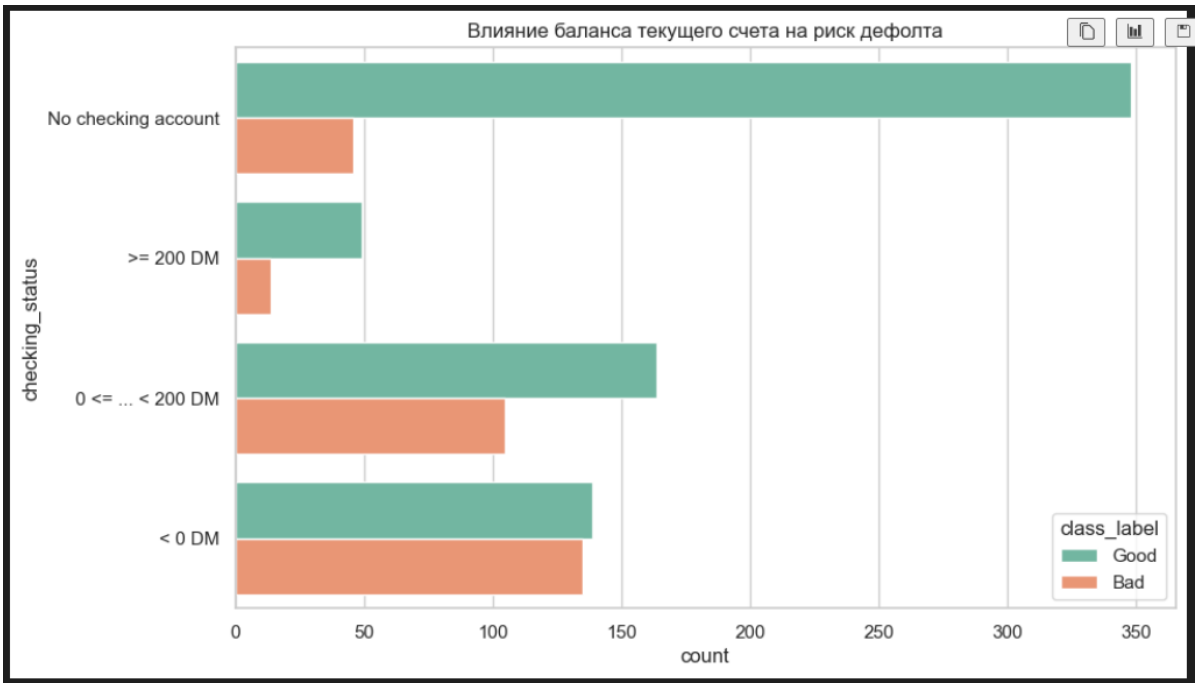

Рис. 1: Зависимость риска от баланса на счете

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.

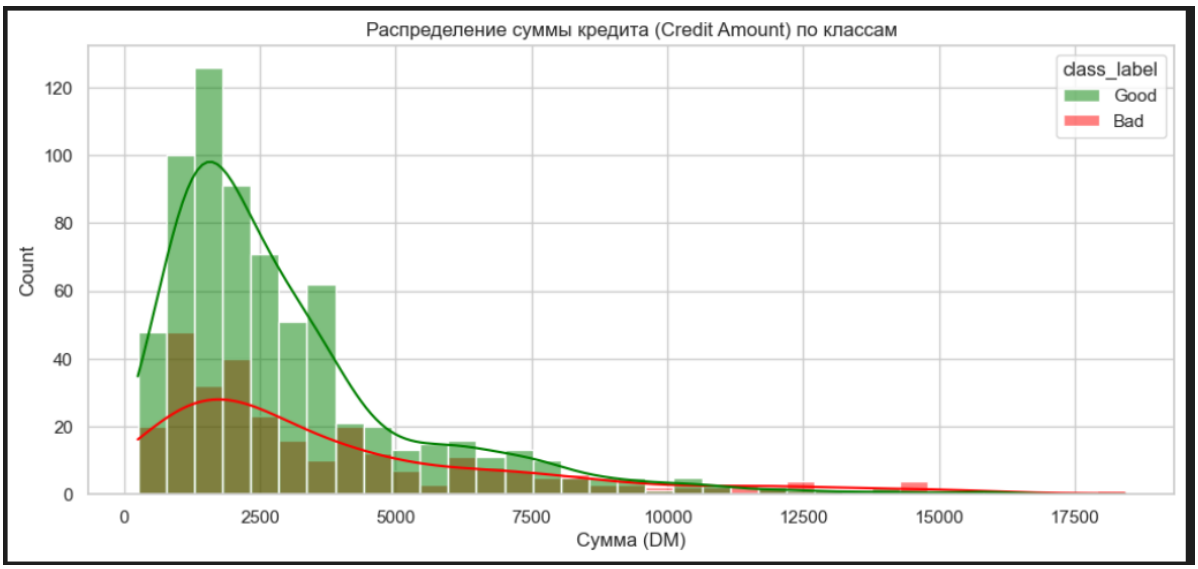

Рис. 2: Распределение суммы кредита (Good vs Bad)

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 числовыми значениями (Label Encoding).

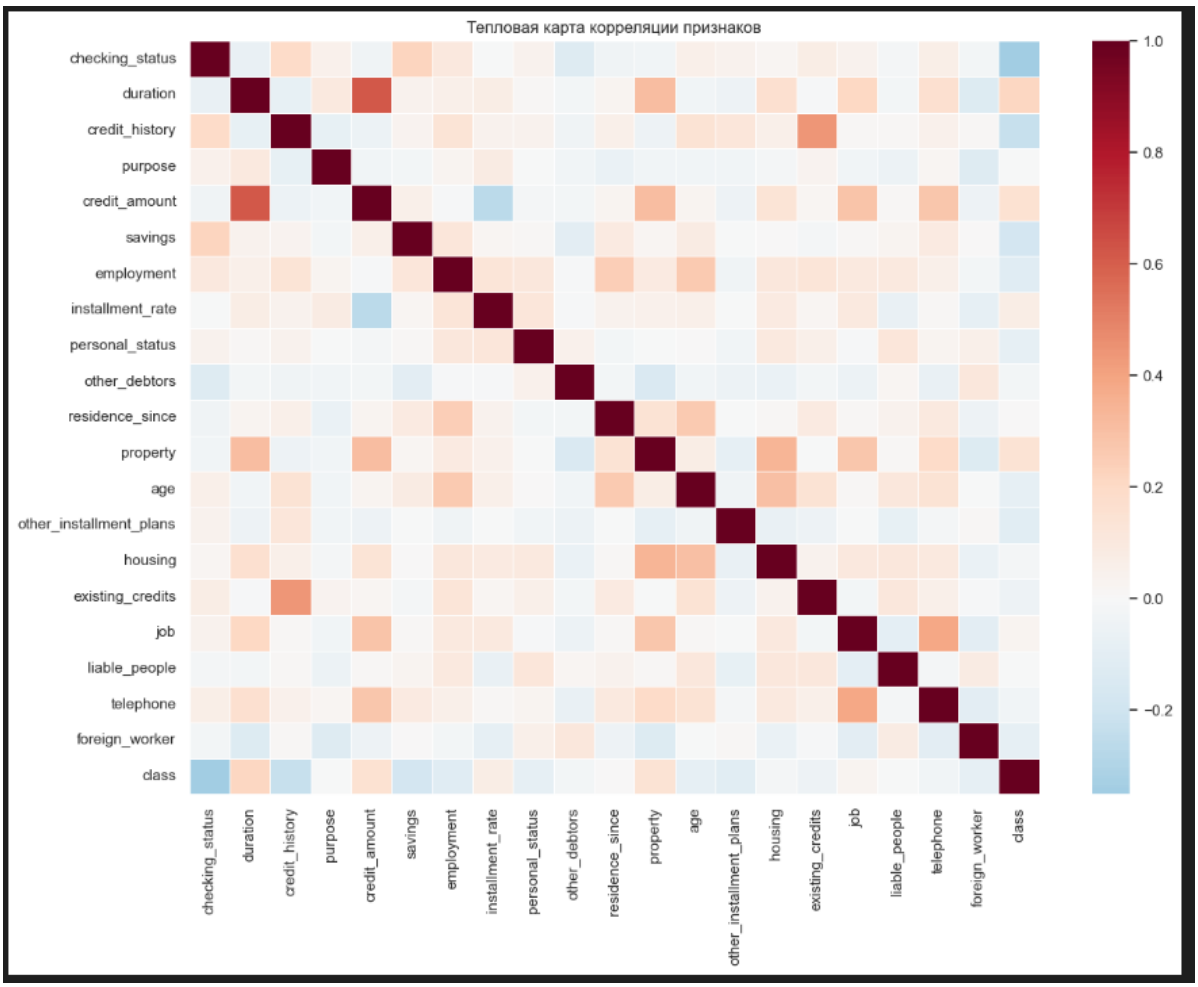

Рис. 3: Тепловая карта корреляций

Интерпретация: Наблюдается **отрицательная корреляция** между целевой переменной `class` и статусом счета `checking_status` (коэф. ≈ -0.35).

Это объясняется кодировкой данных:

- Целевая переменная: 1 = Good, 2 = Bad (чем выше число, тем хуже).
- Статус счета: закодирован от 0 (отрицательный баланс) до 3 (нет счета/надежный).

Также выявлена логичная сильная связь между суммой кредита и его длительностью.

Таким образом, рост финансовой стабильности клиента (увеличение значения атрибута) приводит к снижению вероятности дефолта (уменьшение значения класса к 1). Аналогичная обратная связь прослеживается с кредитной историей и сбережениями.

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.

```sql
SELECT
    purpose, COUNT(*) as total,
    ROUND(AVG(credit_amount), 0) as avg_amount,
    SUM(CASE WHEN risk_status = 'Bad' THEN 1 ELSE 0 END) as bad_count
FROM credits
GROUP BY purpose
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:

1. Самая рискованная цель — **Education** (Образование). Доля невозвратов достигает 44%, несмотря на относительно небольшую среднюю сумму.
2. Кредиты на подержанные автомобили (**Car used**) имеют высокую среднюю сумму (5370 DM), но низкий процент дефолтов (16.5%), что делает их выгодными для банка.
3. 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.

```sql
SELECT age, job, credit_amount, checking_status, purpose
FROM credits
WHERE risk_status = 'Bad' AND age < 35
  AND (checking_status LIKE '%< 0%' OR savings LIKE '%< 100%')
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 рамках данной лабораторной работы был реализован полный конвейер (pipeline) анализа данных: от первичной обработки «сырого» файла до извлечения бизнес-инсайтов с помощью SQL-запросов.

Ниже представлены детализированные выводы по каждому этапу исследования:

6.1 1. Подготовка и предобработка данных

- Успешно загружен датасет *German Credit Data* (1000 записей, 21 атрибут).
- Выявлено отсутствие заголовков в исходном файле, что потребовало ручного назначения имен столбцов согласно документации UCI.
- Проведена проверка качества данных: пропущенные значения (NaN) отсутствуют.
- Выполнено декодирование 13 категориальных признаков (включая `checking_status`, `credit_history`, `purpose`) с использованием словарей. Это позволило перейти от криптографических меток (напр. 'A11', 'A30') к интерпретируемым бизнес-терминам.

6.2 2. Статистический анализ и оценка рисков

- **Дисбаланс классов:** Выборка смещена — 70% надежных клиентов (Good) против 30% дефолтных (Bad). Это означает, что модели машинного обучения потребуют балансировки или использования метрик, отличных от Accuracy.
- **Финансовая оценка:** Рассчитана общая сумма риска. «Плохие» кредиты составляют значительную часть портфеля, при этом средняя сумма дефолтного кредита выше средней суммы по больнице, что подтверждает необходимость внедрения матрицы стоимости (Cost Matrix 5:1).

6.3 3. Визуализация и корреляционный анализ

С помощью библиотек `Seaborn` и `Matplotlib` были выявлены ключевые зависимости:

- **Статус счета (Checking Status):** Является самым сильным индикатором. Клиенты без текущего счета (*No checking account*) парадоксально оказываются самыми надежными, тогда как клиенты с отрицательным балансом (< 0 *DM*) наиболее склонны к дефолту.
- **Распределение сумм:** Гистограммы показали «тяжелый правый хвост» — небольшое количество кредитов на очень крупные суммы часто оказывается невозвратным.
- **Корреляции:** Тепловая карта подтвердила, что наибольшую статистическую связь с целевой переменной имеют признаки состояния счета, кредитной истории и продолжительности кредита.

6.4 4. SQL-аналитика и бизнес-инсайты

Интеграция `Pandas` с `SQLite` позволила провести глубокий сегментированный анализ:

- **Аномалия в целях кредитования:** SQL-агрегация показала, что кредиты на образование (*Education*) являются самыми токсичными (44% невозврата), в то время как кредиты на покупку подержанного авто (*Car used*) — одни из самых безопасных (16.5% риска).
- **Профилирование риска:** С помощью сложного SQL-запроса был выделен сегмент «молодых и рискованных» (возраст < 35 лет, плохая история, отсутствие сбережений). Был идентифицирован конкретный кейс с дефолтом на сумму более 18,000 DM, что указывает на ошибки в текущей скоринговой политике банка для молодых заемщиков.

Итог: Работа показала эффективность совместного использования Python для визуализации и SQL для точечной выборки данных. Полученные результаты могут быть использованы для пересмотра кредитной политики банка, в частности, ужесточения требований к образовательным кредитам и введения дополнительных проверок для молодых клиентов с нулевым балансом на счете.