

Комментарий ревьюера

Привет, Ирина!

Меня зовут Олег Мазуренко, и я буду проверять этот проект. Спасибо за проделанную работу!

Ко мне можно обращаться на «ты». Как мне обращаться, напиши.

Я буду использовать различные цвета, чтобы было удобнее воспринимать мои комментарии:

Синий текст — просто текст комментария.

👉 Зеленый текст — все отлично.

👉 Фиолетовый текст — сделано все правильно, однако есть рекомендации, на что стоит обратить внимание. Реализованные рекомендации позволят нам наработать опыт решения задачи разными способами или посмотреть на задачу под иным углом.

✖ Красный текст — есть недочеты, они иногда бывают.

Любая ошибка это возможность посмотреть на задачу с другой стороны и освоить новые знания, по этому не надо расстраиваться, если они есть.

Обращаю внимание, что комментарии ревьюера после проверки удалять нельзя они помогают отследить логику исправления или изменения проекта. Это дополнительный навык при обучении. Свой исправляемый код лучше тоже не удалять, а ремить #, это позволит акцентировать изменения в ходе подготовки окончательного варианта проекта.

Ответы на мои комментарии лучше тоже помечать. Например: **Комментарий студента**

Теперь посмотрим, что у нас получилось!

Привет. Ко мне можно обращаться на ты.

Комментарий ревьюера

👉 Привет! Хорошо, пошел смотреть проект.

✓ Исследование надежности заемщиков

Комментарий ревьюера 0

👉 Хочу обратить внимание, что во всех следующих проектах необходимо будет делать описание проекта. Где отразить какая цель исследования, какой контекст (особенности отрасли и данных), что планируем делать (план исследования).

В реальном проекте не будет инструкции и задач, которые необходимо выполнить, по этому будем учиться делать описание проекта самостоятельно.

Привет. Ко мне можно обращаться на ты.

Описание проекта

Заказчик — кредитный отдел банка.

Цель исследования - понять есть ли взаимосвязь факта погашения взятого кредита в срок с такими характеристиками заемщика, как:

- наличие детей и их количество
- семейное положение
- ежемесячный доход
- цели кредита: на машину, на недвижимость, на образование, на свадьбу

Входные данные от банка — статистика о платёжеспособности клиентов.

План исследования:

- изучить данные, предоставляемые банком
- предобработка данных:
 - проверить данные на наличие пропусков и выбрать алгоритмы их заполнения или удаления
 - проверить наличие аномальных значений и выбрать алгоритмы их обработки
 - проверить типы данных и привести к соответствующим типам
 - проверить на наличие явных и не явных дубликатов и выбрать алгоритмы из удаления
 - оценить необходимость категоризации данных
- провести исследование зависимости погашения кредита в срок от:
 - количества детей
 - семейного положения
 - уровня дохода
 - цели кредита

- сформулировать выводы на основе полученных результатов исследования

Комментарий ревьюера 2

👉 Все проекты, это как часть реального исследования, за которое деньги заплатят или не заплатят. А отчет читают: сначала описание, о чем идет речь и что делаем, потом выводы: что в итоге получили. А когда у заказчика в голове все сложилось, то смотрят как это делалось. То есть, в описании необходимо отразить какая цель исследования, какой контекст (особенности отрасли и данных), что мы планируем делать (план исследования). И в реальном проекте не будет инструкции и задач, которые необходимо выполнить, по этому следует научиться делать описание проекта самостоятельно.

Во второй части проекта вы выполните шаги 3 и 4. Их вручную проверит ревьюер. Чтобы вам не пришлось писать код заново для шагов 1 и 2, мы добавили авторские решения в ячейки с кодом.

Откройте таблицу и изучите общую информацию о данных

Задание 1. Импортируйте библиотеку pandas. Считайте данные из csv-файла в датафрейм и сохраните в переменную data. Путь к файлу:

/datasets/data.csv

```
import pandas as pd

try:
    data = pd.read_csv('/datasets/data.csv')
except:
    data = pd.read_csv('https://code.s3.yandex.net/datasets/data.csv')
```

Задание 2. Выведите первые 20 строчек датафрейма data на экран.

data.head(20)

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	debt	total_income
0	1	-8437.673028	42	высшее	0	женат / замужем	0	F	сотрудник	0	25386
1	1	-4024.803754	36	среднее	1	женат / замужем	0	F	сотрудник	0	11206
2	0	-5623.422610	33	Среднее	1	женат / замужем	0	M	сотрудник	0	14586
3	3	-4124.747207	32	среднее	1	женат / замужем	0	M	сотрудник	0	26766
4	0	340266.072047	53	среднее	1	гражданский брак	1	F	пенсионер	0	15866
5	0	-926.185831	27	высшее	0	гражданский брак	1	M	компаньон	0	25576
6	0	-2879.202052	43	высшее	0	женат / замужем	0	F	компаньон	0	24056
7	0	-152.779569	50	СРЕДНЕЕ	1	женат / замужем	0	M	сотрудник	0	13586
8	2	-6929.865299	35	ВЫСШЕЕ	0	гражданский брак	1	F	сотрудник	0	9586
9	0	-2188.756445	41	среднее	1	женат / замужем	0	M	сотрудник	0	14446
10	2	-4171.483647	36	высшее	0	женат / замужем	0	M	компаньон	0	11396
11	0	-792.701887	40	среднее	1	женат / замужем	0	F	сотрудник	0	7706

гражданский

Задание 3. Выведите основную информацию о датафрейме с помощью метода info().

```
data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21525 entries, 0 to 21524
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   children              21525 non-null  int64  
 1   days_employed         19351 non-null  float64
 2   dob_years             21525 non-null  int64  
 3   education             21525 non-null  object  
 4   education_id          21525 non-null  int64  
 5   family_status         21525 non-null  object  
 6   family_status_id      21525 non-null  int64  
 7   gender                21525 non-null  object  
 8   income_type           21525 non-null  object  
 9   debt                  21525 non-null  int64  
10   total_income          19351 non-null  float64
11   purpose               21525 non-null  object  
dtypes: float64(2), int64(5), object(5)
memory usage: 2.0+ MB

```

✓ Предобработка данных

✓ Удаление пропусков

Задание 4. Выведите количество пропущенных значений для каждого столбца. Используйте комбинацию двух методов.

```
data.isna().sum()
```

```

children          0
days_employed    2174
dob_years         0
education         0
education_id      0
family_status     0
family_status_id  0
gender            0
income_type       0
debt              0
total_income      2174
purpose           0
dtype: int64

```

Задание 5. В двух столбцах есть пропущенные значения. Один из них — `days_employed`. Пропуски в этом столбце вы обработаете на следующем этапе. Другой столбец с пропущенными значениями — `total_income` — хранит данные о доходах. На сумму дохода сильнее всего влияет тип занятости, поэтому заполнить пропуски в этом столбце нужно медианным значением по каждому типу из столбца `income_type`. Например, у человека с типом занятости `сотрудник` пропуск в столбце `total_income` должен быть заполнен медианным доходом среди всех записей с тем же типом.

```

for t in data['income_type'].unique():
    data.loc[(data['income_type'] == t) & (data['total_income'].isna()), 'total_income'] = \
    data.loc[(data['income_type'] == t), 'total_income'].median()

```

✓ Обработка аномальных значений

Задание 6. В данных могут встречаться артефакты (аномалии) — значения, которые не отражают действительность и появились по какой-то ошибке. таким артефактом будет отрицательное количество дней трудового стажа в столбце `days_employed`. Для реальных данных это нормально. Обработайте значения в этом столбце: замените все отрицательные значения положительными с помощью метода `abs()`.

```
data['days_employed'] = data['days_employed'].abs()
```

Задание 7. Для каждого типа занятости выведите медианное значение трудового стажа `days_employed` в днях.

```
data.groupby('income_type')['days_employed'].agg('median')
```

```

income_type
безработный    366413.652744
в декрете      3296.759962
госслужащий    2689.368353
компаньон      1547.382223
пенсионер      365213.306266

```

```
предприниматель    520.848083
сотрудник           1574.202821
студент             578.751554
Name: days_employed, dtype: float64
```

У двух типов (безработные и пенсионеры) получатся аномально большие значения. Исправить такие значения сложно, поэтому оставьте их как есть.

Задание 8. Выведите перечень уникальных значений столбца children .

```
data['children'].unique()

array([ 1,  0,  3,  2, -1,  4, 20,  5])
```

Задание 9. В столбце children есть два аномальных значения. Удалите строки, в которых встречаются такие аномальные значения из датафрейма data .

```
data = data[(data['children'] != -1) & (data['children'] != 20)]
```

Задание 10. Ещё раз выведите перечень уникальных значений столбца children , чтобы убедиться, что артефакты удалены.

```
data['children'].unique()

array([1, 0, 3, 2, 4, 5])
```

✓ Удаление пропусков (продолжение)

Задание 11. Заполните пропуски в столбце days_employed медианными значениями по каждому типу занятости income_type .

```
for t in data['income_type'].unique():
    data.loc[(data['income_type'] == t) & (data['days_employed'].isna()), 'days_employed'] = \
        data.loc[(data['income_type'] == t), 'days_employed'].median()
```

Задание 12. Убедитесь, что все пропуски заполнены. Проверьте себя и ещё раз выведите количество пропущенных значений для каждого столбца с помощью двух методов.

```
data.isna().sum()

children          0
days_employed    0
dob_years         0
education         0
education_id      0
family_status     0
family_status_id  0
gender            0
income_type       0
debt              0
total_income      0
purpose           0
dtype: int64
```

✓ Изменение типов данных

Задание 13. Замените вещественный тип данных в столбце total_income на целочисленный с помощью метода astype() .

```
data['total_income'] = data['total_income'].astype(int)
```

✓ Обработка дубликатов

Задание 14. Обработайте неявные дубликаты в столбце education . В этом столбце есть одни и те же значения, но записанные по-разному: с использованием заглавных и строчных букв. Приведите их к нижнему регистру.

```
data['education'] = data['education'].str.lower()
```

Задание 15. Выведите на экран количество строк-дубликатов в данных. Если такие строки присутствуют, удалите их.

```
data.duplicated().sum()
```

```
71
```

```
data = data.drop_duplicates()
```

✓ Категоризация данных

Задание 16. На основании диапазонов, указанных ниже, создайте в датафрейме data столбец total_income_category с категориями:

- 0–30000 — 'E';
- 30001–50000 — 'D';
- 50001–200000 — 'C';
- 200001–1000000 — 'B';
- 1000001 и выше — 'A'.

Например, кредитополучателю с доходом 25000 нужно назначить категорию 'E', а клиенту, получающему 235000, — 'B'.

Используйте собственную функцию с именем categorize_income() и метод apply().

```
def categorize_income(income):
    try:
        if 0 <= income <= 30000:
            return 'E'
        elif 30001 <= income <= 50000:
            return 'D'
        elif 50001 <= income <= 200000:
            return 'C'
        elif 200001 <= income <= 1000000:
            return 'B'
        elif income >= 1000001:
            return 'A'
    except:
        pass
```

```
data['total_income_category'] = data['total_income'].apply(categorize_income)
```

Задание 17. Выведите на экран перечень уникальных целей взятия кредита из столбца purpose.

```
data['purpose'].unique()
```

```
array(['покупка жилья', 'приобретение автомобиля',
      'дополнительное образование', 'сыграть свадьбу',
      'операции с жильем', 'образование', 'на проведение свадьбы',
      'покупка жилья для семьи', 'покупка недвижимости',
      'покупка коммерческой недвижимости', 'покупка жилой недвижимости',
      'строительство собственной недвижимости', 'недвижимость',
      'строительство недвижимости', 'на покупку подержанного автомобиля',
      'на покупку своего автомобиля',
      'операции с коммерческой недвижимостью',
      'строительство жилой недвижимости', 'жилье',
      'операции со своей недвижимостью', 'автомобили',
      'заняться образованием', 'сделка с подержанным автомобилем',
      'получение образования', 'автомобиль', 'свадьба',
      'получение дополнительного образования', 'покупка своего жилья',
      'операции с недвижимостью', 'получение высшего образования',
      'свой автомобиль', 'сделка с автомобилем',
      'профильное образование', 'высшее образование',
      'покупка жилья для сдачи', 'на покупку автомобиля', 'ремонт жилья',
      'заняться высшим образованием'], dtype=object)
```

Задание 18. Создайте функцию, которая на основании данных из столбца purpose сформирует новый столбец purpose_category, в который войдут следующие категории:

- 'операции с автомобилем',
- 'операции с недвижимостью',
- 'проведение свадьбы',
- 'получение образования'.

Например, если в столбце purpose находится подстрока 'на покупку автомобиля', то в столбце purpose_category должна появиться строка 'операции с автомобилем'.

Используйте собственную функцию с именем `categorize_purpose()` и метод `apply()`. Изучите данные в столбце `purpose` и определите, какие подстроки помогут вам правильно определить категорию.

```
def categorize_purpose(row):
    try:
        if 'автом' in row:
            return 'операции с автомобилем'
        elif 'жил' in row or 'недвиж' in row:
            return 'операции с недвижимостью'
        elif 'свад' in row:
            return 'проведение свадьбы'
        elif 'образов' in row:
            return 'получение образования'
    except:
        return 'нет категории'

data['purpose_category'] = data['purpose'].apply(categorize_purpose)
```

✓ Шаг 3. Исследуйте данные и ответьте на вопросы

✓ 3.1 Есть ли зависимость между количеством детей и возвратом кредита в срок?

Посчитаем количество задолженностей по кредитам сгруппированные по кол-ву детей у заемщика

```
debts_groupby_children = data.groupby('children')['debt'].sum()
debts_groupby_children
```

```
children
0    1063
1     444
2     194
3       27
4        4
5         0
Name: debt, dtype: int64
```

На первый взгляд самая большая задолженность у заемщиков без детей.

Сгруппируем задолженности по двум категориям:

- задолженности заемщиков без детей
- задолженности заемщиков с любым количеством детей

```
debts_without_children = debts_groupby_children[0]
print('Задолженности заемщиков без детей:', debts_without_children)
debts_with_children = debts_groupby_children.sum() - debts_without_children
print('Задолженности заемщиков с детьми:', debts_with_children)
```

```
Задолженности заемщиков без детей: 1063
Задолженности заемщиков с детьми: 669
```

Опять получается, что задолженностей заемщиков без детей больше.

Посчитаем относительное число задолженностей. Ведь количество взятый кредитов в группах может сильно отличаться и это число сильно влияет на количество просроченных возвратов кредитов.

Для этого сначала рассчитаем количество всех взятых кредитов и сгруппируем по количеству детей.

```
count_credits = data.groupby('children')['children'].count()
count_credits
```

```
children
0    14091
1     4808
2     2052
3       330
4        41
5         9
Name: children, dtype: int64
```

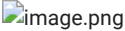
Действительно, количество взятых кредитов заемщиками без детей сильно больше, чем другими заемщиками, поэтому стоит рассчитать относительное число задолженностей в каждой категории.

Рассчитаем относительное число задолженностей к общему числу кредитов в зависимости от числа детей и отсортируем по убыванию

```
del_debt = debts_groupby_children*100/count_credits
print('Относительное число задолженностей к общему числу кредитов в зависимости от числа детей:', del_debt.sort_values(ascending = False))
```

```
↗ Относительное число задолженностей к общему числу кредитов в зависимости от числа детей: children
4    9.756098
2    9.454191
1    9.234609
3    8.181818
0    7.543822
5    0.000000
dtype: float64
```

Комментарий ревьюера 0

👉 Лучше строить сводную таблицу со следующими колонками: 'Всего кредитополучателей', 'Всего должников', 'Доля должников'. В этом случае таблица строится со следующими параметрами: `aggfunc=['count', 'sum', 'mean']`. Примерно вот так: 

Правильно ли я понимаю, что во всех заданиях лучше делать как в коде ниже?

Сгруппируем задолженности по количеству детей и создадим сводную таблицу с данными: "Всего кредитополучателей", "Всего должников", "Доля должников"

```
#создание сводной таблицы с агрегирующей функцией рассчитывающей число кредитов, число задолженностей, долю задолженностей
#по столбцу debt - наличие задолженности
data_pivot_children = data.pivot_table(index='children', values='debt', aggfunc=['count', 'sum', 'mean'])
data_pivot_children.columns = ['Всего кредитополучателей', 'Всего должников', 'Доля должников']

#сбросить multiindex использовать первую часть индекса
#data_pivot_children.columns = data_pivot_children.columns.droplevel(1)
#data_pivot_children = data_pivot_children.rename(
#    columns={
#        'count': 'Всего кредитополучателей',
#        'sum': 'Всего должников',
#        'mean': 'Доля должников',
#    }
#)
data_pivot_children.style.format("{:.3f}")
```

```
↗
```

	Всего кредитополучателей	Всего должников	Доля должников
children			
0	14091.000	1063.000	0.075
1	4808.000	444.000	0.092
2	2052.000	194.000	0.095
3	330.000	27.000	0.082
4	41.000	4.000	0.098
5	9.000	0.000	0.000

Картина изменилась. Наибольшая относительная частота задолженностей у заемщиков с 4 детьми, а у заемщиков без детей наименьший процент относительных задолженностей.

Комментарий ревьюера

👉 Можно сделать вот так, через функцию. О, сначала нужно было проколотаться и самой написать код, а потом увидеть, что Олег уже написал мне пример! Я долго не могла понять, как избавиться от Мультииндекса. Мои попытки оставила внутри закоментированного кода. Так тоже работало. Крутая идея объединить всё в одну функцию и в качестве аргумента передавать столбец, по которому проводить исследование, т.к. во всех исследованиях порядок действий аналогичен! Спасибо. Буду использовать в будущем.

Комментарий ревьюера 2

👉 Короткий код - показатель профессионализма. Мы профи! Или как? 😊

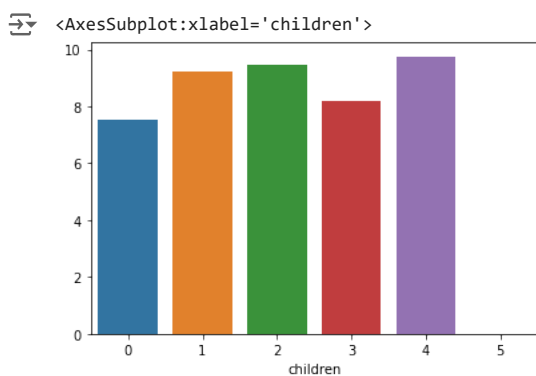
```
# Комментарий ревьюера
# Создаем копию датасета, что бы не работать с исходным
temp = data.copy()
# Напишем функцию, так как все задачи аналогичны
def que(category):
    data_temp = temp.pivot_table(index=category, values='debt', aggfunc=['count', 'sum', 'mean'])
    data_temp.columns = ['Всего кредитополучателей', 'Всего должников', 'Доля должников']
    # Оформим таблицу цветным градиентом, но можно ее вывести и просто display(data_temp)
    display(data_temp.style.format("{:.3f}").background_gradient(cmap='Blues', axis=0))

que('children')
```

	Всего кредитополучателей	Всего должников	Доля должников
children			
0	14091.000	1063.000	0.075
1	4808.000	444.000	0.092
2	2052.000	194.000	0.095
3	330.000	27.000	0.082
4	41.000	4.000	0.098
5	9.000	0.000	0.000

```
import seaborn
```

```
seaborn.barplot(x=del_debt.index, y=del_debt)
```



Комментарий ревьюера 0

👉 Обращаю внимание, что диаграммы и графики должны содержать подписи осей и название графика на русском языке. В следующих проектах это будет критической ошибкой.

Попробую ниже с помощью кода изменить подписи к графикам

```
import seaborn as sns
```

```
#создание столбчатой диаграммы
```

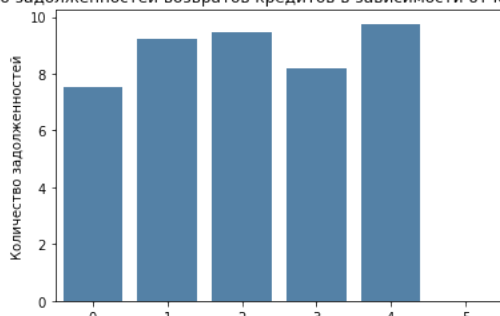
```
ax = sns.barplot (x=del_debt.index, y=del_debt,
    color='steelblue')
```

```
#задание подписей к диаграмме
```

```
ax.set (xlabel='Количество детей',
    ylabel='Количество задолженностей',
    title='Количество задолженностей возвратов кредитов в зависимости от количества детей')
ax
```

```
<AxesSubplot:title={'center':'Количество задолженностей возвратов кредитов в зависимости от количества детей'}, xlabel='Количество детей', ylabel='Количество задолженностей'>
```

Количество задолженностей возвратов кредитов в зависимости от количества детей



Комментарий ревьюера 2

👉 Красивые и информативные графики всегда приветствуются, особенно когда на них есть все подписи осей, названия меток и название графика. Как правило, заказчик не является аналитиком и ему трудно по коду разбираться, что изображено на диаграммах. Кроме того, в презентацию реального проекта в основном идут именно скрины графиков, таблицы мало информативны. По этому необходимо сразу графики максимально описывать.

Из диаграммы видно, что минимальная частота задолженностей у заемщиков без детей. У заемщиков с одним, двумя и четырьмя детьми частота задолженностей примерно одинакова в районе 9-9,8%. Выборка по заемщикам с 5 детьми не показательна, т.к. в ней всего 9 заемщиков. Так же видно, что у заемщиков с тремя детьми так же задолженность случается реже, чем у других заемщиков с детьми. Но этот вывод не показателен, т.к. выборка заемщиков с 3 детьми состоит из 330 чел, против заемщиков без детей 14091 - выборки отличаются в 42 раза. Следовательно лучше посмотреть разницу между заемщиками с детьми и заемщиками без детей, не учитывая количества детей.

Сгруппируем взятые кредиты по двум категориям:

- количество кредитов заемщиков без детей
- количество кредитов заемщиков с любым количеством детей

```
count_credit_without_children = count_credits[0]
print('Количество взятых кредитов заемщиками без детей:', count_credit_without_children)
count_credit_with_children = count_credits.sum() - count_credits[0]
print('Количество взятых кредитов заемщиками с любым количеством детей:', count_credit_with_children)
print('Отличие количества заемщиков с детьми и без детей:', count_credit_without_children/count_credit_with_children)
```

```
→ Количество взятых кредитов заемщиками без детей: 14091
   Количество взятых кредитов заемщиками с любым количеством детей: 7240
   Отличие количества заемщиков с детьми и без детей: 1.9462707182320442
```

Комментарий ревьюера 0

✗ Ну вот, ошибка кода. Вынужден прервать проверку. 🙄 

Ну вот, переименовала переменную и не заметила, что это отразилось на остальных данных

Из расчета выше видно, что выборки заемщиков без детей и с детьми отличаются в два раза.

Рассчитаем относительные задолженности для двух категорий

```
del_debt_without_children = debts_without_children * 100 / count_credit_without_children
print('Количество относительных задолженностей заемщиков без детей:', del_debt_without_children)
del_debt_with_children = debts_with_children * 100 / count_credit_with_children
print('Количество относительных задолженностей заемщиков с детьми:', del_debt_with_children)
print('Разница между заемщиками без детей и с детьми составляет, %:', del_debt_with_children - del_debt_without_children)
```

```
→ Количество относительных задолженностей заемщиков без детей: 7.543822297920658
   Количество относительных задолженностей заемщиков с детьми: 9.240331491712707
   Разница между заемщиками без детей и с детьми составляет, %: 1.696509193792049
```

Вывод: Задолженностей по возврату кредита в срок у заемщиков без детей меньше (7,5 %), чем у заемщиков с детьми (9,2%).

Разница между ними 1,7%. Опириую этими данными, нужно помнить, что размеры выборок отличаются в два раза (14091>7240).

Если бы выборки были примерно одинаковы, то данные были бы более показательны.

Комментарий ревьюера

👉 Правильный вывод, действительно семьи без детей не несут дополнительных расходов и своевременно обслуживают свой кредит. У клиентов с детьми более высокая расходная часть семейного бюджета, поэтому и возникают сложности со своевременной выплатой по кредитным обязательствам.

Хорошо, что было обращено внимание на несбалансированность выборок, при малой выборке нельзя делать обоснованные выводы.

👉 При сравнении долей (а здесь не равные выборки), необходимо учитывать не разницу в долях, а процент, когда максимальная доля принимается за 100%. В данном случае разница между крайними значениями долей должников (не учитывая категорию с 5-тью детьми) составляет более 22%.

Поняла. 100% - это 9,8% задолженности с 4мя детьми, 7,5% - это $7,5 \cdot 100\% / 9,8 = 76,5\%$ - задолженностей без детей и разница между ними $100\% - 76,5\% = 23,5\%$

Комментарий ревьюера 2

👉 Правильно. Если бы у нас были одинаковые выборки, то тогда могли бы сравнивать доли.

✓ 3.2 Есть ли зависимость между семейным положением и возвратом кредита в срок?

```
debt_family_status = data.groupby('family_status')['debt'].sum().sort_values(ascending = False)
credits_family_status = data.groupby('family_status')['debt'].count().sort_values(ascending = False)
dels_family_status = (debt_family_status*100/credits_family_status).sort_values(ascending = False)
print('Просроченные возвраты кредита в зависимости от семейного положения: ', debt_family_status)
print()
print('Всего взятых кредитов в зависимости от семейного положения:', credits_family_status)
print()
print('Относительное число задолженностей к общему числу кредитов в зависимости от семейного положения:', dels_family_status)
print()
print('Общее число всех кредитов', count_credits.sum())
```

↗ Просроченные возвраты кредита в зависимости от семейного положения: family_status

женат / замужем	927
гражданский брак	385
Не женат / не замужем	273
в разводе	84
вдовец / вдова	63

Name: debt, dtype: int64

Всего взятых кредитов в зависимости от семейного положения: family_status

женат / замужем	12261
гражданский брак	4134
Не женат / не замужем	2796
в разводе	1189
вдовец / вдова	951

Name: debt, dtype: int64

Относительное число задолженностей к общему числу кредитов в зависимости от семейного положения: family_status

Не женат / не замужем	9.763948
гражданский брак	9.313014
женат / замужем	7.560558
в разводе	7.064760
вдовец / вдова	6.624606

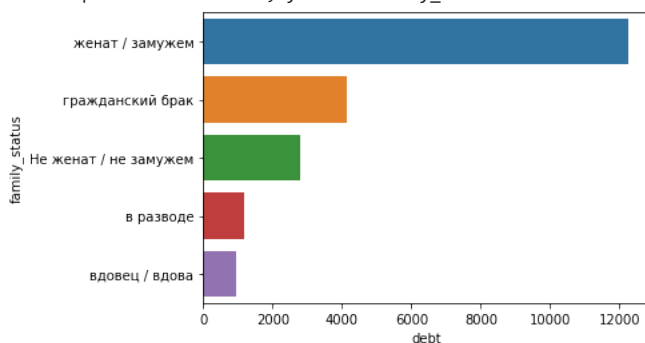
Name: debt, dtype: float64

Общее число всех кредитов 21331

Построим диаграмму для числа взятых кредитов в зависимости от семейного положения

```
seaborn.barplot(x=credits_family_status, y= credits_family_status.index)
```

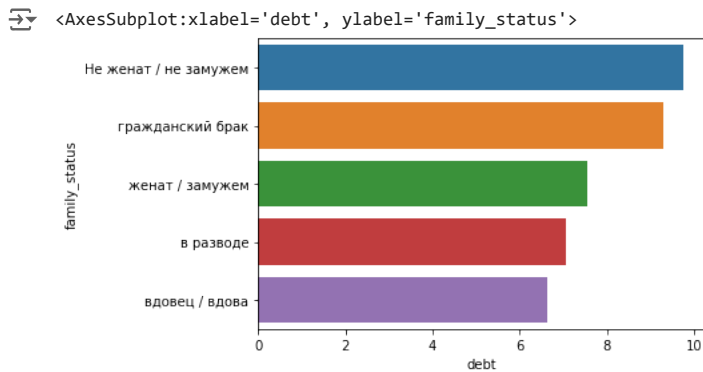
↗ <AxesSubplot:xlabel='debt', ylabel='family_status'>



Из диаграммы видно, что число людей, состоящих в браке (12261) берет кредиты существенно чаще, чем остальные категории граждан.

Построим диаграмму, показывающую относительное количество задолженностей в зависимости от семейного положения.

```
seaborn.barplot(x= dels_family_status , y= dels_family_status.index)
```



Из диаграммы не ясно как семейное положение влияет на количество задолженностей.

Данных по людям состоящим в браке существенно больше, поэтому нужно разбить данные о взятых кредитах на две категории:

- заемщики состоящие в браке
- все остальные

```
count_credits_family = credits_family_status['женат / замужем']
count_credits_single = count_credits.sum() - count_credits_family
print('Заемщики состоящие в браке:', count_credits_family)
print('Заемщики не состоящие в браке:', count_credits_single)
```

```
Заемщики состоящие в браке: 12261
Заемщики не состоящие в браке: 9070
```

Посчитаем количество задолженностей заемщиков в браке и всех остальных:

```
debts_family = debts_family_status['женат / замужем']
debts_single = debts_family_status.sum() - debts_family
print('Количество задолженностей заемщиков состоящих в браке:', debts_family)
print('Количество задолженностей заемщиков не состоящих в браке:', debts_single)
```

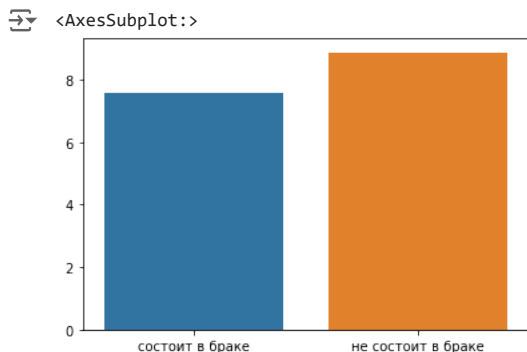
```
Количество задолженностей заемщиков состоящих в браке: 927
Количество задолженностей заемщиков не состоящих в браке: 805
```

Посчитаем относительное количество задолженностей для каждой группы

```
del_family = debts_family * 100 / count_credits_family
del_single = debts_single * 100 / count_credits_single
print('Относительное количество задолженностей заемщиков состоящих в браке:', del_family)
print('Относительное количество задолженностей заемщиков не состоящих в браке:', del_single)
```

```
Относительное количество задолженностей заемщиков состоящих в браке: 7.560557866405676
Относительное количество задолженностей заемщиков не состоящих в браке: 8.875413450937156
```

```
seaborn.barplot(x= ['состоит в браке', 'не состоит в браке'], y= [del_family, del_single])
```



Вывод:

Задолженностей по возврату кредита в срок у заемщиков состоящих в браке меньше (7,6%), чем у остальных категорий граждан (8,9%): "гражданский брак", "Не женат / не замужем", "в разводе", "вдовец / вдова" Разница составляет 1,3%

Комментарий ревьюера

✖ Не корректный вывод, меньше всего задолжность у вдовцов. Анализ необходимо проводить по категориям банка.

Вывод: Из диаграммы долей задолженностей от семейного положения видно, что наименьшая задолженность у вдовцов/вдов - 6,6%, тогда как у людей не состоящих в браке доля задолженностей 9,8% приближается к 10%. Это можно объяснить тем, что люди не состоящие в браке - это чаще молодые неопытные люди, чаще идущие на риск, не учитывающие, что могут возникнуть в жизни непредвиденные обстоятельства. Вдовцами же чаще становятся люди в возрасте уже более мудрые, опытные и более осторожные. В целом данные косвенно свидетельствуют в зависимости задолженностей от возраста и с ростом возраста количество задолженностей падает. По данным из датасета эту гипотезу можно проверить разбив возраста по категориям.

Комментарий ревьюера 2

👉 Оформленные семейные отношения или их прошлый опыт налагают определенную ответственность и приучают к финансовой дисциплине.

✓ 3.3 Есть ли зависимость между уровнем дохода и возвратом кредита в срок?

```
debts_income_ct = data.groupby('total_income_category')['debt'].sum()
print('Просроченные возвраты кредита в зависимости от уровня дохода: ', debts_income_ct)
print()
count_income_ct = data.groupby('total_income_category')['total_income_category'].count()
print('Количество взятых кредитов в зависимости от уровня дохода: ', count_income_ct)
print()
del_income_ct = debts_income_ct * 100 / count_income_ct
print('Относительное число просроченных возвратов кредита в зависимости от уровня дохода: ', del_income_ct)
```

```
➡ Просроченные возвраты кредита в зависимости от уровня дохода: total_income_category
A      2
B     354
C    1353
D      21
E        2
Name: debt, dtype: int64
```

```
Количество взятых кредитов в зависимости от уровня дохода: total_income_category
A      25
B     5014
C    15921
D      349
E       22
Name: total_income_category, dtype: int64
```

```
Относительное число просроченных возвратов кредита в зависимости от уровня дохода: total_income_category
A      8.000000
B     7.060231
C     8.498210
D     6.017192
E     9.090909
dtype: float64
```

- 0–30000 — 'E';
- 30001–50000 — 'D';
- 50001–200000 — 'C';
- 200001–1000000 — 'B';
- 1000001 и выше — 'A'.

```
income_categorys = ['1000001 и выше :A', '200001–1000000 :B', '50001–200000 :C', '30001–50000 :D', '0–30000 :E ']
debts_income_ct.index
```

```
➡ Index(['A', 'B', 'C', 'D', 'E'], dtype='object', name='total_income_category')
```

Диаграмма просроченные возвраты кредита в зависимости от уровня дохода

```
seaborn.barplot(x = debts_income_ct , y= income_categorys)
```

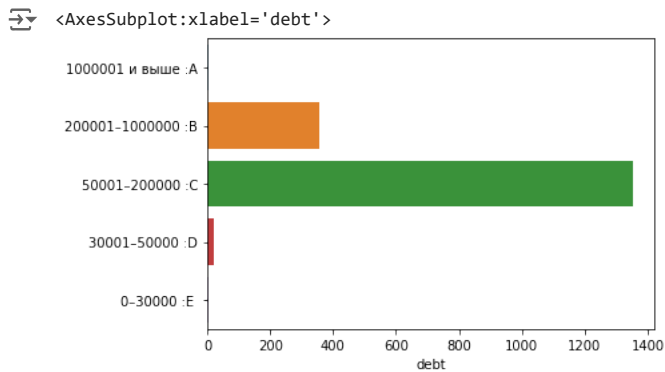


Диаграмма взятых кредитов в зависимости от уровня дохода

```
seaborn.barplot(x = count_income_ct , y= income_categorys)
```

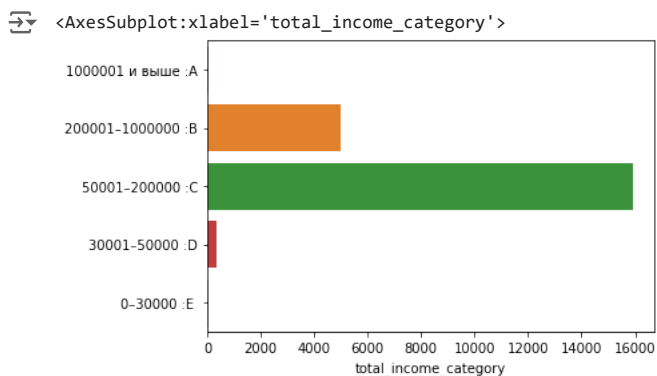
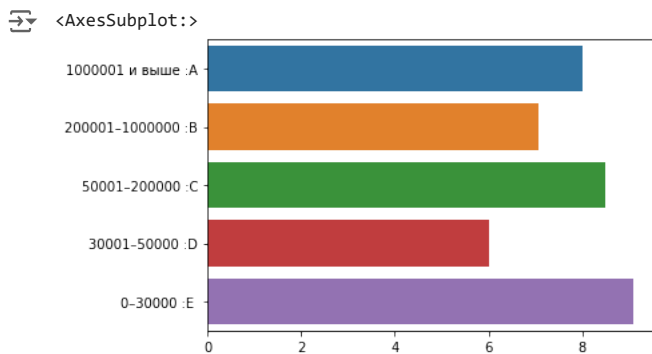


Диаграмма относительного числа просроченных возвратов кредита в зависимости от уровня дохода

```
seaborn.barplot(x = del_income_ct , y= income_categorys)
```



Категории уровня дохода А (более млн) и Е (менее 30тыс) крайне малочисленны. В них 25 и 22 человека. Они выбиваются из всей выборки и их можно не учитывать, они дают не показательные результаты. Самый большой процент задолженностей 8,5% по возврату кредитов всрок в категории С (50001-200000). Это так же самая многочисленная группа из всех. В ней чаще всего берут кредиты и чаще всего не возвращают в срок. В группе В (200001 - 1000000) с более высоким заработком процент задолженности 7% снижается. В группе D (30001–50000) с более низким доходом процент задолженности 6% самый маленький. Это так же и самая малочисленная группа. Т.е. люди из этой группы реже берут кредиты, но если берут, то более аккуратно относятся к погашению кредита.

Вывод:

Прослеживается зависимость возвратов кредитов в срок с уровнем дохода. Чаще всего берут кредиты люди из категории С и в этой же категории чаще всего не возвращают кредиты в срок 8 % людей из категории. В категории D самый маленький процент задолженностей по позврату кредита 6%. С ростом уровня дохода свыше 200000 процент задолженностей падает до 7%

Комментарий ревьюера

👍 Хорошо, что обратили внимание на то, что представленные выборки несбалансированы и какой-то вывод можно делать только по двум категориям с достаточным размером выборок.

👉 В задачах, где требуется анализировать несбалансированные выборки можно, делать дополнительный анализ, разбив данные на

равные выборки методом .qcut . Подробнее можно прочитать тут:
<https://dfedorov.spb.ru/pandas/Разделение%20данных%20в%20Pandas%20с%20помощью%20qcut%20и%20cut.html>.

Комментарий ревьюера 2

👉 Можно сделать цикл с перебором количества категорий.

```
# Комментарий ревьюера 2
temp = data.copy()
start, stop = 3, 8 # начальное и конечное количество категорий
while start <= stop:
    temp['new_category'] = pd.qcut(temp['total_income'], q=start, precision=0)
    que('new_category')
    start += 1
```

	Всего кредитополучателей	Всего должников	Доля должников
new_category			
(20666.0, 119218.0]	7111.000	580.000	0.082
(119218.0, 172357.0]	7238.000	627.000	0.087
(172357.0, 2265604.0]	6982.000	525.000	0.075
Всего кредитополучателей			
Всего должников			
Доля должников			
new_category			
(20666.0, 107507.0]	5333.000	427.000	0.080
(107507.0, 142594.0]	5450.000	480.000	0.088
(142594.0, 195842.0]	5216.000	444.000	0.085
(195842.0, 2265604.0]	5332.000	381.000	0.071
Всего кредитополучателей			
Всего должников			
Доля должников			
new_category			
(20666.0, 98514.0]	4267.000	344.000	0.081
(98514.0, 132113.0]	4266.000	358.000	0.084
(132113.0, 161380.0]	4266.000	373.000	0.087
(161380.0, 214604.0]	4266.000	358.000	0.084
(214604.0, 2265604.0]	4266.000	299.000	0.070
Всего кредитополучателей			
Всего должников			
Доля должников			
new_category			
(20666.0, 92092.0]	3556.000	285.000	0.080
(92092.0, 119218.0]	3555.000	295.000	0.083
(119218.0, 142594.0]	3672.000	327.000	0.089
(142594.0, 172357.0]	3566.000	300.000	0.084
(172357.0, 228893.0]	3427.000	276.000	0.081
(228893.0, 2265604.0]	3555.000	249.000	0.070
Всего кредитополучателей			
Всего должников			
Доля должников			
new_category			
(20666.0, 87287.0]	3048.000	233.000	0.076
(87287.0, 113563.0]	3047.000	262.000	0.086
(113563.0, 137465.0]	3047.000	263.000	0.086
(137465.0, 155314.0]	3047.000	264.000	0.087
(155314.0, 184554.0]	3047.000	257.000	0.084
(184554.0, 242009.0]	3047.000	242.000	0.079
(242009.0, 2265604.0]	3048.000	211.000	0.069
Всего кредитополучателей			
Всего должников			
Доля должников			
new_category			
(20666.0, 83837.0]	2667.000	206.000	0.077
(83837.0, 107507.0]	2666.000	221.000	0.083
(107507.0, 127546.0]	2666.000	233.000	0.087
(127546.0, 142594.0]	2784.000	247.000	0.089
(142594.0, 166525.0]	2549.000	223.000	0.087
(166525.0, 195842.0]	2667.000	221.000	0.083
(195842.0, 254250.0]	2665.000	196.000	0.074
(254250.0, 2265604.0]	2667.000	185.000	0.069

Комментарий ревьюера 2

👉 Вот, нашлась самая кредито-опасная категория дохода заемщиков, повод для более глубокого анализа.

3.4 Как разные цели кредита влияют на его возврат в срок?

```
# Ваш код будет здесь. Вы можете создавать новые ячейки.
debt_purpose = data.groupby('purpose_category')['debt'].sum()
print('Задолженности от цели:', debt_purpose)
print()
count_purpose = data.groupby('purpose_category')['purpose_category'].count()
print('Количество кредитов от цели:', count_purpose)
```

```
print()
del_purpose = debt_purpose*100/count_purpose
print('Процент задолженностей от цели:', del_purpose.sort_values(ascending = False))
```

```
Задолженности от цели: purpose_category
операции с автомобилем      400
операции с недвижимостью    780
получение образования       369
проведение свадьбы          183
Name: debt, dtype: int64
```

```
Количество кредитов от цели: purpose_category
операции с автомобилем      4279
операции с недвижимостью   10751
получение образования       3988
проведение свадьбы          2313
Name: purpose_category, dtype: int64
```

```
Процент задолженностей от цели: purpose_category
операции с автомобилем      9.347978
получение образования       9.252758
проведение свадьбы          7.911803
операции с недвижимостью    7.255139
dtype: float64
```

Вывод:

Прослеживается зависимость задолженностей по возврату кредита в срок от целей кредита. Самый большой процент задолженности на операции с автомобилем и на получение образования 9,3%. Самый низкий процент задолженностей на операции с недвижимостью 7,3%. Средний процент - на проведение свадеб 7,9%.

Комментарий ревьюера

Правильно, видимо, кредиты на автомобиль и обучение несут дополнительные риски связанные или с возможной аварией, или с проблемами трудоустройства после обучения.

3.5 Приведите возможные причины появления пропусков в исходных данных.

Ответ:

технические причины: сбой при записи таблиц, файлов, данных. Человеческий фактор: ошибки при заполнении, перезабытии данных, усталость.

Комментарий ревьюера

Существует несколько причин отсутствия данных в датасетах, в том числе:

- Человеческий фактор: Ошибки ввода данных, нежелание отвечать на определенные вопросы.
- Технические проблемы: Возможны сбои с оборудованием или программным обеспечением для сбора данных. Отсутствующие данные могут быть удалены в процессе обработки или очистки данных.
- Организационные: Конфиденциальность, могут отсутствовать данные, идентифицирующие отдельных лиц или компании.

3.6 Объясните, почему заполнить пропуски медианным значением — лучшее решение для количественных переменных.

Ответ: среднее вычисляется по всем данным, и единичные резкие скачки в данных могут сильно повлиять на значение среднего арифметического, на медианное значение такие скачки не оказывают влияния, поэтому его лучше использовать для количественных переменных.

Комментарий ревьюера

Правильно, у средних более высокая чувствительность к выбросам, чем у медиан. Можно попробовать посчитать и сравнить среднюю и медиану для колонки с выбросами, например `days_employed`.

Шаг 4: общий вывод.

В целом наблюдается некоторая связь между возвратами кредитов в срок и количеством детей, семейным положением, целями на которые берется кредит. Но из-за неравномерности выборок, разной их численности возникает сомнение в статистической значимости полученных результатов.

Комментарий ревьюера

✗ Общий вывод относится ко всему проекту, а не только к исследовательскому анализу.

Он должен содержать развернутое резюме по всем разделам проекта: описание данных, порядок обработки аномалий, пропусков и дубликатов, предобработка данных для анализа и проведенный анализ.

В реальных проектах сначала читают цель исследования, затем полученные выводы, затем уже сам анализ.

👉 В общем выводе можно дать рекомендации по улучшению сбора данных на основе обнаруженных ошибок. А так же составить портрет добросовестного заемщика.

Ниже новый общий вывод. Его не буду выделять цветом.

Анализ данных, предоставленных банком

Банком предоставлена статистика по заемщикам, содержащая следующие данные:

- children — количество детей в семье
- days_employed — общий трудовой стаж в днях
- dob_years — возраст клиента в годах
- education — уровень образования клиента
- education_id — идентификатор уровня образования
- family_status — семейное положение
- family_status_id — идентификатор семейного положения
- gender — пол клиента
- income_type — тип занятости
- debt — имел ли задолженность по возврату кредитов
- total_income — ежемесячный доход
- purpose — цель получения кредита

Объем данных составил: 21525 записей

В данных наблюдаются:

- пропуски значений в столбцах days_employed — общий трудовой стаж и total_income - тип занятости
- отрицательные значения в столбце children — количество детей в семье
- данные в разных регистрах в purpose — цель получения кредита
- столбец total_income — ежемесячный доход заполнен данными с незначащими числами после запятой.
- одни и те же данные в столбце purpose — цель получения кредита сформулированы по-разному

Вывод по предварительному анализу данных: необходима предобработка данных до проведения исследования, включающая поиск и устранение пропусков данных, артефактов данных (отрицательные значения, значения не внушающие доверие), привести типы данных, категоризовать данные. Необходимо провести анализ на наличие явных и скрытых дубликатов и удалить их.

Предобработка данных

В данных days_employed и total_income пропуски в 2174 ячейках соответственно.

Пропуски в данных days_employed и total_income были заполнены медианными значениями рассчитанными отдельно для каждого типа занятости income_type

В данных days_employed отрицательные значения заменены на значения по модулю

Строки с аномальными значениями в children (количество детей в семье) -1 и 20 удалены

Тип данных в total_income изменен на целочисленный, т.к. значения после запятой не несут в данных суммах ежемесячного дохода существенного значения.

Данные в столбце education приведены к единому нижнему регистру.

Анализ показал наличие 71 дублирующийся строк, которые были удалены.

Для проведения исследования взаимосвязи задолженностей по кредитам с ежемесячным доходом все суммы разбиты на 4 категории дохода в зависимости от суммы

Данные столбца- цель кредита разбиты на 4 категории: операции с автомобилем, недвижимостью, проведение свадьбы, получение образования

Вывод по предобработке данных и рекомендации по сбору данных: Предоставленные банком данные требовали предварительной обработки. В данных встречались пропуски значений, аномалии в значениях, такие как отрицательные значения в стаже и количестве детей, встречались данные не вызывающие доверия, различные формулировки одних и тех же понятий, данные не были поделены на категории.

При сборе данных необходимо обратить внимание на следующие моменты: • Исключить запись отрицательных значений в столбцы с числовыми значениями, в которых отрицательных значений быть не может • Записывать данные в одном регистре • Заранее

разделять данные на категории, где это возможно • Не допускать различных формулировок одного и того же понятия • По возможности не допускать пропусков в данных • Следить, чтобы информация не вносилась повторно, чтобы не было дублей, одинаковых строк с данными

Исследование данных

В исследовательской части была проведена работа по выявлению зависимости возврата кредита в срок от: • количества детей • семейного положения • уровня дохода • цели кредита

В ходе исследования были рассчитаны следующие данные для каждого вида исследования: • Всего кредитополучателей • Всего должников • Доля должников Рассчитанные данные были сгруппированы соответственно по количеству детей, по семейному положению, уровню дохода и цели кредита.

Исследование зависимости между возвратом кредита в срок и количеством детей

В ходе исследования установлено, что минимальная частота задолженностей у заемщиков без детей - 7,5%

У заемщиков с одним, двумя и четырьмя детьми частота задолженностей примерно одинакова в районе 9,2-9,8%.

Выборка по заемщикам с 5 детьми не показательна, т.к. в ней всего 9 заемщиков, против 14091 заемщиков без детей.

У заемщиков с тремя детьми так же задолженность случается реже, чем у других заемщиков с детьми – 8,2%.

Но утверждать, что семьи с тремя детьми более надежные по сравнению с семьями с 2 или 4 детьми нельзя, т.к. все выборки сильно не сбалансированы по количеству заемщиков.

В ходе исследования принято решение посчитать разницу между семьями с детьми и без детей. Вывод остался тот же, задолженности у заемщиков без детей случаются реже, в 7,5% случаях. У заемщиков с детьми задолженности случаются чаще, в 9,2% случаях.

Такой вывод и не удивителен, т.к. при наличии в семье детей возникают разные непредвиденные и неучтенные расходы, связанные с детьми, что может помешать вернуть кредит в срок.

Исследование зависимости между возвратом кредита в срок и семейным положением

В ходе исследования установлено, что наименьшая задолженность у вдовцов/вдов - 6,6%,

У людей не состоящих в браке доля задолженностей 9,8%.

Это можно объяснить тем, что люди не состоящие в браке - это чаще молодые неопытные люди, чаще идущие на риск, не учитывающие, что могут возникнуть в жизни непредвиденные обстоятельства. Вдовцами же чаще становятся люди в возрасте уже более мудрые, опытные и более осторожные.

Исследование зависимости между возвратом кредита в срок и уровнем дохода

Для проведения исследования взаимосвязи задолженностей по кредитам с ежемесячным доходом все суммы доходов разбиты на категории: • 0–30000 — 'E'; • 30001–50000 — 'D'; • 50001–200000 — 'C'; • 200001–1000000 — 'B'; • 1000001 и выше — 'A'.

Установлено, что заемщики из категории C чаще всего не возвращают кредиты в срок – в 8,5% случаях.

В категории D самый маленький процент задолженностей по возврату кредита - 6%, но выборка не многочисленна (349 чел), поэтому можно учитывать результаты только по двум самым многочисленным выборкам, по выборке C (15921 чел) и по выборке B (5014 чел)

С ростом уровня дохода, категория B, процент задолженностей падает до 7%.

Люди из категории C чаще всего берут кредиты и чаще всего их не возвращают в срок. С ростом дохода свыше 200000 у заемщиков увеличивается возможность гасить обязательства по кредиту в срок. А с уменьшением дохода ниже 50000 люди более осторожно относятся к самому факту взятия на себя обязательств по кредиту.

Частоту задолженностей для категорий A и E не следует учитывать в исследовании совсем в связи с крайне немногочисленными выборками: 25 и 22 кредитополучателя соответственно.

Исследование зависимости между возвратом кредита в срок и целью кредита

Для проведения исследования взаимосвязи задолженностей с целью кредита все цели были разбиты на 4 категории: • 'операции с автомобилем', • 'операции с недвижимостью', • 'проведение свадьбы', • 'получение образования'.

Самый большой процент задолженности на операции с автомобилем и на получение образования 9,3% Самый низкий процент задолженностей на операции с недвижимостью 7,3% Средний процент - на проведение свадеб 7,9%

Чтобы объяснить данные цифры желательно провести дополнительные исследования, например исследование от суммы кредита, т.к. сумма кредита на недвижимость несравнима с суммами на остальные цели.

Выводы по всем исследованиям

По проведенным исследованиям можно составить портрет самого добросовестного кредитоплательщика: это человек без детей, вдовец или вдова с уровнем дохода 200001–1000000 берущий кредит на операции с недвижимостью.

Портрет же самого ненадежного кредитополучателя: человек с детьми, не состоящий в браке, с уровнем дохода 50001–200000 берущий кредит на покупку автомобиля и на получение образования.

По результатам исследования стало понятно, что необходимо провести дополнительные исследования: • Так как многие выборки получились неравномерными необходимо проводить исследования с помощью методов учитывающих неравномерность выборок • Провести исследование зависимости задолженностей от возраста заемщика • Провести исследования зависимости задолженности от суммы кредита • Разбить категорию уровень ежемесячного дохода С на более мелкие категории и провести исследования в этих категориях

В целом наблюдается некоторая связь между возвратами кредитов в срок и количеством детей, семейным положением, целями на которые берется кредит. Но из-за неравномерности выборок, разной их численности возникает сомнение в статистической значимости полученных результатов.

Что-то получилось слишком много текста, наверное я переборщила

Комментарий ревьюера 2

👍 Хорошо, теперь правильно. 👍

Заключительный комментарий ревьюера 0

👍 Ирина!

✖ Увы, должен прервать ревью из-за ошибки в коде. Настоятельно рекомендую перед отправкой проекта на ревью проверять в JupyterHub исполнение кода тетрадки.

Жду исправленный проект. Если будут трудности, можно обратиться к куратору.

Заключительный комментарий ревьюера

👍