САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ – ПРОЦЕССОВ УПРАВЛЕНИЯ
КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ЭНЕРГЕТИЧЕСКИХ СИСТЕМ

Татарченкова Анна Дмитриевна

Магистерская диссертация
Моделирование страхования кредитных рисков

Направление 010402 «Прикладная математика и информатика»
Основная образовательная программа ВМ.5505.2019 «Математическое и информационное обеспечение экономической деятельности»

Научный руководитель:
доктор физ.-мат. наук,
профессор
Смирнов Н.В.

Санкт-Петербург
2021

Содержание

Введение

На современном этапе развития экономики важную роль играет кредитование. Многие происходящие сегодня процессы без него будут совершенно невозможны. Банки, выдавая кредиты, неизбежно несут потери, с ними связанные. Основным риском в кредитовании является невозврат заемных средств, поэтому банки применяют различные стратегии и методы для сокращения кредитных рисков. Один из таких методов – это страхование.

Проблема страхования кредитных рисков в наше время очень актуальна, поскольку кредиты берутся всеми и повсеместно, но дать точную оценку возможных потерь может быть достаточно сложно. Страхуя риск, связанный с невозвратом кредита, банк частично или полностью перекладывает потери по данному договору на страховую компанию, минимизируя таким образом собственные потери.

Механизмы страхования кредитных рисков на сегодняшний день все ещё недостаточно развиты, поскольку сама система страхования кредитов появилась сравнительно недавно. Сейчас лишь немногие банки в России имеют развитую схему страхования кредитных рисков.

Данная работа посвящена теме страхования кредитных рисков.

Целью научно-исследовательской работы является анализ проблемы оценки и страхования кредитных рисков.

Для осуществления поставленной цели необходимо решить следующие задачи:

1. Изучить основные понятия актуарной и финансовой математики, которые будут использоваться в дальнейшем для построения модели страхования кредита;

2. Построить модель, позволяющую вычислить стоимость страхования кредита;

3. Оценить компоненты построенной модели при помощи моделей актуарной математики, известных в страховании жизни;

4. Получить оценку компонент модели на основе банковских статистических данных и вычислить предполагаемую стоимость страхования кредита;

5. Модифицировать полученную модель для случая, когда учитываются такие риски как кризис и неплатежеспособность заемщика, и вычислить предполагаемую стоимость страхования кредита в этих условиях.

Обзор литературы

Сегодня на тему страхования существует множество научных статей, книг, а также учебных пособий.

Однако тема кредитного страхования жизни рассматривается в достаточно небольшом количестве научных источников. Среди них можно выделить работу Е.А. Беляевских [2], в которой была построена, подробно изучена и исследована модель страхования кредитных рисков.

Понятия актуарной математики из области страхования жизни достаточно подробно описаны в [1].

Основные математические модели и методы, которые необходимы для определения характеристик продолжительности жизни, страховых надбавок, интенсивности отказов и т.д. можно найти в книге [6].

Материал, который обеспечивает понимание принципов работы сферы страхования, методов расчета финансового обеспечения продуктов страховых компаний представлен в [3], [7]. Также заслуживают внимания статьи [9] и [10], в которых можно найти более подробное описание кредитного страхования и страховых схем в банковской системе. В источнике [15] представлены таблицы продолжительности жизни, необходимые при вычислении актуарной приведенной стоимости страхования кредита.

Данные, необходимые для численных расчетов на основе банковской статистики, можно найти на сайте [12]. Источник [8] представляет собой сборник статей, в которых находятся формулы расчета энтропии, которые были применены для вычисления рейтинга заемщика. В [5] автор описывает алгоритм расчета рейтинга платежеспособности заемщика для выдачи кредита. Работа [4] посвящена моделированию некоторых вероятностных характеристик в целях системного изучения и статистического анализа продолжительности жизни населения.

Алгоритм для вычисления страховой премии на основании расчета рейтинга неплатежеспособности основан на построении классификатора – дерева решений. Информация об общих принципах, терминологии, структуре,

процессе построения, преимуществах и областях применения можно найти на сайтах [11], [13]. Калькулятор расчета страховой премии реализуется посредством языка программирования Python, в котором используется пакет прикладных математических процедур SciPy, необходимых для обработки данных. Информацию о нем можно найти в источнике [14].

Общая постановка задачи

Общую постановку задачи можно разделить на следующие этапы:

1. Построение модели кредитного страхования с различной функцией интенсивности отказов при выплате заемщиком ежегодно или m раз в год;
2. Оценка компонентов модели с дискретными и непрерывными выплатами;
3. Вычисление предполагаемой стоимости страхования кредита;
4. Модификация модели за счет выявление дополнительных параметров, влияющих на расчет предполагаемой стоимости;
5. Построение алгоритма и его реализация для расчета страховой премии на основании вычисления рейтинга неплатежеспособности заемщика.

§1. Описание модели

Здесь будет построена исходная актуарная модель стоимости страхования кредита, а затем рассмотрены различные вариации компонентов.

1.1. Общая модель расчета стоимости страхования кредита

Сформулируем основные предположения, необходимые для построения модели, следуя работе [2]:

- Заемщик получает в банке кредит и должен выплатить назад сумму в размере C. Сумма выплачивается за n итераций, по $\frac{C}{n}$ за один раз;
- Банковская ставка равна i процентам за определенный промежуток времени, например, год. Следовательно, по окончании -го года заемщик выплатит банку сумму в размере $\frac{C}{n}(1+i)^n$;
- Таким образом, финансовая приведенная стоимость кредита

$$C_n = \frac{C}{n}((1+i) + (1+i)^2 + \cdots + (1+i)^n) =$$

$$= \frac{C}{n}(1+i)\frac{(1+i)^n - 1}{i} = \frac{C}{n}\frac{(1+i)^n - 1}{d},$$

  где $d = \frac{i}{1+i} -$ учетная ставка;
- Пусть заемщик выплачивал долг банку за k итераций, $k \in [1, n-1]$, но не смог погасить задолженность по кредиту до конца. В таких случаях, чтобы гарантированно не потерять свои средства, банк может заключить договор со страховой компанией, чтобы она выплатила банку сумму, которую не смог выплатить клиент;
- Страховая компания выплатит банку

$$C_k = \frac{C}{n}((1+i)^{k+1} + \cdots + (1+i)^n) = \frac{C}{n}(1+i)^{k+1}\frac{(1+i)^{n-k} - 1}{i} =$$

$$= \frac{C}{n}\frac{(1+i)^{n+1} - (1+i)^{k+1}}{i};$$

- Вероятность того, что кредит, который выплачивали x периодов, будут выплачивать еще по крайней мере k периодов, обозначим ${}_kp_x$.

Аналогично, ${}_kq_x$ – вероятность того, что выплаты по кредиту прекратятся в течение первых k периодов;

- Тогда ${}_kp_xq_{x+k}$ – вероятность того, что кредит «возраста» x будет выплачиваться еще k периодов, а потом выплаты по нему прекратятся;
- Пусть интенсивность отказов или интенсивность смертности $\mu(x)$

$$\mu(x) = \frac{-s\prime(x)}{s(x)},$$

функция, которая для каждого возраста x дает значение в точке x случайной величины X при условии дожития до возраста x. Где $s(x) -$ функция дожития, которая для любого $x > 0$ есть вероятность того, что кредит будет выплачиваться до момента x. Установим зависимость между ${}_kp_x$ и $\mu(x)$. Для этого заменим x на y и проинтегрируем

$$-\mu(y)dy = dlns(y)$$

$$-\int_x^{x+k} \mu(y)dy = ln\frac{s(x+k)}{s(x)} = ln\ {}_kp_x$$

Откуда получим, что

$${}_kp_x = e^{\left[-\int_x^{x+k}\mu(y)dy\right]}. \quad (1)$$

Тогда

$$q_{x+k} = 1 - e^{\left[-\int_{x+k}^{x+k+1}\mu(y)dy\right]}; \quad (2)$$

- С учетом дисконтирования на момент $k+1$ получаем актуарную приведенную стоимость страхования кредита

$$A_C = \sum_{k=1}^{n} C_k v^{k+1}\ {}_kp_xq_{x+k} = \frac{C}{n}\sum_{k=1}^{n}\frac{(1+i)^{n-k}-1}{i}\ {}_kp_xq_{x+k},$$

где $v = \frac{1}{1+i} -$ коэффициент дисконтирования.

В случае непрерывных выплат, то есть, таким образом, чтобы дисконтирование производилось непрерывно, за бесконечно малые промежутки времени

$$A_C = \frac{C}{(e^{\delta}-1)n}\int_1^n (e^{\delta})^{n-t}\ {}_kp_xq_{x+t}dt,$$

где $\delta -$ непрерывная процентная ставка.

1.2. Случай m-кратных выплат

Предположим, что выплаты по кредиту происходят с периодичностью m раз в год в течение n лет, то есть, предполагается nm выплат по $\frac{C}{nm}$ каждая. Пусть новая процентная ставка $i^{(m)} -$ номинальная процентная ставка при $m -$ кратном конвертировании. Таким образом, если кредит выплачивается m раз в год равными частями, то сумма платежа составит

$$\frac{C(1+i)^n}{nm} = \frac{C}{nm} + Ci^{(m)}.$$

Также предположим, что кредит выплачивался в течение j периодов, $j \in [1, nm-1]$, а затем выплаты прекратились. На основании этого

$$C_j = \frac{C}{nm}(1+i^{(m)})^{j+1}\frac{(1+i^{(m)})^{nm-j}-1}{i^{(m)}},$$

$$A_C^{(m)} = \frac{C}{i^{(m)}nm}\sum_{j=1}^{nm}\left[(1+i^{(m)})^{nm-j}-1\right]\ {}_{\frac{j}{m}}p_x\,{}_{\frac{1}{m}}q_{x+\frac{j}{m}},$$

где $j = km + \frac{s}{m}$, $k -$ количество полных лет, $s -$ остаток, например, количество месяцев.

§2. Расчет актуарной приведенной стоимости страхования кредита с различными моделями оценки интенсивности отказов

Будем рассматривать полученную актуарную приведенную стоимость в рамках различных моделей актуарной математики.

2.1. Случай с постоянной интенсивностью отказов

В формуле актуарной приведенной стоимости страхования кредита присутствуют компоненты ${}_kp_x$ и q_{x+k}, значения которых зависят от выбора модели их оценки. Эти величины зависят от интенсивности отказов $\mu(x)$, оценить которую можно по разному.

Пусть $\mu(x) -$ постоянная $\mu(x) = const$ [1]. Обратимся к формулам (1) и (2). Откуда

$$\int_x^{x+k} \mu(y)dy = \mu(x+k-x) = \mu k.$$

Следовательно,

$$_kp_x = e^{-\mu k},$$
$$q_{x+k} = 1 - e^{-\mu}.$$

Таким образом,

$$A_C = \frac{C}{n}\sum_{k=1}^{n} \frac{(1+i)^{n-k}-1}{i}\ {}_kp_x q_{x+k} =$$
$$= \frac{C}{n}\sum_{k=1}^{n} \frac{(1+i)^{n-k}-1}{i} e^{-\mu k}(1-e^{-\mu}) =$$
$$= \frac{C}{n}(1-e^{-\mu})\frac{1}{i}\left[\sum_{k=1}^{n}(1+i)^{n-k}e^{-\mu k} - \sum_{k=1}^{n} e^{-\mu k}\right] =$$

$$= \frac{C}{ni}(1-e^{-\mu})\left[\frac{[(1+i)e^{\mu}]^n-1}{[(1+i)e^{\mu}-1]e^{\mu n}} - e^{-\mu}\frac{1-e^{-\mu n}}{1-e^{-\mu}}\right];$$

Если же проценты начисляются непрерывно, учитывая $1+i = e^{\delta}$,

$$A_C = \frac{C}{ni}(1-e^{-\mu})\int_1^n((1+i)^{n-t}-1)e^{-\mu t}dt\ = \frac{C}{n(e^{\delta}-1)}(1-$$
$$-e^{-\mu})\int_1^n\left((e^{\delta})^{n-t}-1\right)e^{-\mu t}dt = \frac{C}{n(e^{\delta}-1)}(1-e^{-\mu})\left[\int_1^n\left((e^{\delta n-\delta t-\mu t}\right)dt\ -\right.$$
$$\left.-\int_1^n e^{-\mu t}\ dt\right] = \frac{C}{n}\frac{(1-e^{-\mu})}{(1-e^{\delta})}\left[\frac{e^{-\mu}-e^{-\mu n}}{\mu} - \frac{e^{\delta(n-1)-\mu}-e^{-\mu n}}{\delta+\mu}\right].$$

2.2. Случай с интенсивностью отказов, выраженной моделью де Муавра

В модели, предложенной Абрахамом де Муавром, интенсивность отказов приближается функцией $\mu(x) = \frac{1}{\omega-x}$ [6], где $x-$ количество периодов с уже произведенными выплатами по кредиту, $\omega-$ некоторая константа – предельное количество периодов.

В этой модели,

$$-\int_x^{x+k}\frac{1}{\omega-y}dy = -(-\ln(\omega-y))\,|_x^{x+k} = ln\frac{\omega-x-k}{\omega-x} = \ln\left(1-\frac{k}{\omega-x}\right);$$

$$-\int_{x+k}^{x+k+1}\frac{1}{\omega-y}dy=\ln(1-\frac{1}{\omega-x-k});$$

Откуда

$$_kp_x=1-\frac{k}{\omega-x},$$

$$q_{x+k}=\frac{1}{\omega-x-k}.$$

Учитывая полученные значения, актуарная приведенная стоимость страхования кредита

$$A_C=\frac{C}{n}\sum_{k=1}^{n}\frac{(1+i)^{n-k}-1}{i}\ {}_kp_xq_{x+k}=\frac{C}{n}\sum_{k=1}^{n}\frac{(1+i)^{n-k}-1}{i}(1-\frac{k}{\omega-x})(\frac{1}{\omega-x-k})=$$

$$\frac{C}{n}\sum_{k=1}^{n}\frac{(1+i)^{n-k}-1}{i}\left(\frac{1}{\omega-x}\right)=\frac{C}{ni}\frac{1}{\omega-x}\left[\frac{(1+i)^n-1}{i}-n\right].$$

Для непрерывного начисления процентов, при следующей замене с непрерывной процентной ставкой $1+i=e^{\delta}$ получим

$$A_C=\frac{C}{ni}\frac{1}{\omega-x}\int_1^n[(1+i)^{n-t}-1]dt=\frac{C}{n(e^{\delta}-1)}\frac{1}{\omega-x}\int_1^n\left[e^{\delta n-\delta t}-\right.$$

$$\left.-1\right]dt=\frac{C}{n(e^{\delta}-1)}\frac{1}{\omega-x}\left[\frac{e^{\delta n-\delta t}}{-\delta}|_1^n-t|_1^n\right]=\frac{C}{n(e^{\delta}-1)}\frac{1}{\omega-x}\left[\frac{e^{\delta(n-1)}-1}{\delta}-n+1\right].$$

2.3. Случай с интенсивностью отказов, выраженной моделью Мэйкхема

Мейкхем предложил приближать интенсивность отказов функцией вида $\mu(x)=A+Be^{\alpha x}$[6], где постоянное слагаемое A позволяет учесть риски для жизни, связанные с несчастными случаями, а член $Be^{\alpha x}$ учитывает влияние возраста на смертность.

Рассмотрим $\mu(x)=A+Be^{\alpha x}$

$$\int_x^{x+k}(A+Be^{\alpha y})dy=Ay|_x^{x+k}+\frac{Be^{\alpha y}}{\alpha}|_x^{x+k}=Ak+\frac{Be^{\alpha x}(e^{\alpha k}-1)}{\alpha},$$

$$\int_{x+k}^{x+k+1}(A+Be^{\alpha y})dy=A+\frac{Be^{\alpha x}\left(e^{\alpha k}-1\right)}{\alpha}).$$

Получим

$$_kp_x=e^{-(Ak+\frac{Be^{\alpha x}\left(e^{\alpha k}-1\right)}{\alpha})},$$

$$q_{x+k}=1-e^{-(A+\frac{Be^{\alpha(x+k)}(e^{\alpha}-1)}{\alpha})}.$$

Таким образом,

$$A_C = \frac{C}{ni}\sum_{k=1}^{n}[(1+i)^{n-k}-1]\left[e^{-(Ak+\frac{Be^{\alpha x}\left(e^{\alpha k}-1\right)}{\alpha})} - e^{-(A(k+1)+\frac{Be^{\alpha x}\left(e^{\alpha(k+1)}-1\right)}{\alpha})}\right];$$

В случае, когда выплаты происходят непрерывно получим

$$A_C = \frac{C}{n(e^{\delta}-1)}\int_1^n \left[\left[(e^{\delta})^{n-t}-1\right]\left[e^{-(At+\frac{Be^{\alpha x}(e^{\alpha t}-1)}{\alpha})} - e^{-(A(t+1)+\frac{Be^{\alpha x}\left(e^{\alpha(t+1)}-1\right)}{\alpha})}\right]\right] dt.$$

§3. Использование таблицы продолжительности жизни

В страховании жизни часто можно встретить таблицы продолжительности жизни людей определенного пола и национальности. Рассмотрим таблицу 1, в которой использованы данные о демографической ситуации мужского населения в 2016 году [15]. Здесь указаны данные о количестве людей, доживших до указанного возраста. Рассматриваемая таблица отражает данные для возрастов, кратных пяти, для краткости. Будем предполагать, что в рамках промежутка $[5n, 5(n+1)]$, где $n = 0, 1, \dots, 19$, сохраняется количество людей, которые дожили до соответствующего возраста.

Таблица 1 – Таблица продолжительности жизни мужского населения в России

Возраст x	lx
0	100000
5	98205
10	97950
15	97682
20	96843
25	94996
30	92217
35	88837
40	84473
45	78627
50	71139
55	61893
60	51795
65	40626
70	29895
75	19680
80	11132
85	5179

90	1679
95	340
100	58

На основании этих данных будем производить вычисления в контексте выплаты кредитов. Будем считать, что $l_x -$ это количество кредитов, которые продолжают выплачиваться к моменту x со времени их выдачи. Поэтому вероятности ${}_kp_x$ и ${}_kq_x$ будут вычисляться следующим образом

$$ {}_kp_x = \frac{l_{x+k}}{l_x},$$

$$ {}_kq_x = \frac{l_{x+k}-l_{x+k+1}}{l_{x+k}},$$

а их произведение

$$ {}_kp_xq_{x+k} = \frac{l_{x+k}-l_{x+k+1}}{l_x}.$$

Полученные значения будем использовать при вычислении актуарной приведенной стоимости страхования кредита по формулам, приведенным ранее.

§4. Расчеты с использованием банковской статистики

Рассмотрим банковскую статистику неплатежей по кредитам [12], имеющим определенные характеристики, и построим на ее основе модель, описывающую интенсивность отказов (объем невыплат) в зависимости от «времени жизни» кредита.

4.1. Общий случай

Чтобы иметь возможность вести статистику и оценивать возможные потери, все кредиты группируют на основе определенных критериев. Такими критериями являются сумма кредита (диапазон значений), процентная ставка, срок кредита, периодичность платежей и время выдачи. Кредиты, имеющие одинаковые характеристики относят к одной группе и называют поколением.

Были проанализированные данные по 46 поколениям потребительских кредитов сроком 4 года с ежемесячными выплатами. Все вычисления сделаны в пакете Excel. Поскольку все кредиты в поколении примерно одинаковы в денежном измерении, примем суммы кредитования равными.

Данные о неплатежах в графическом представлении выглядят следующим образом, где на оси абсцисс представлена временная шкала, на оси ординат – доля новых невыплат по кредиту сроком более 90 дней (см. рис. 1).

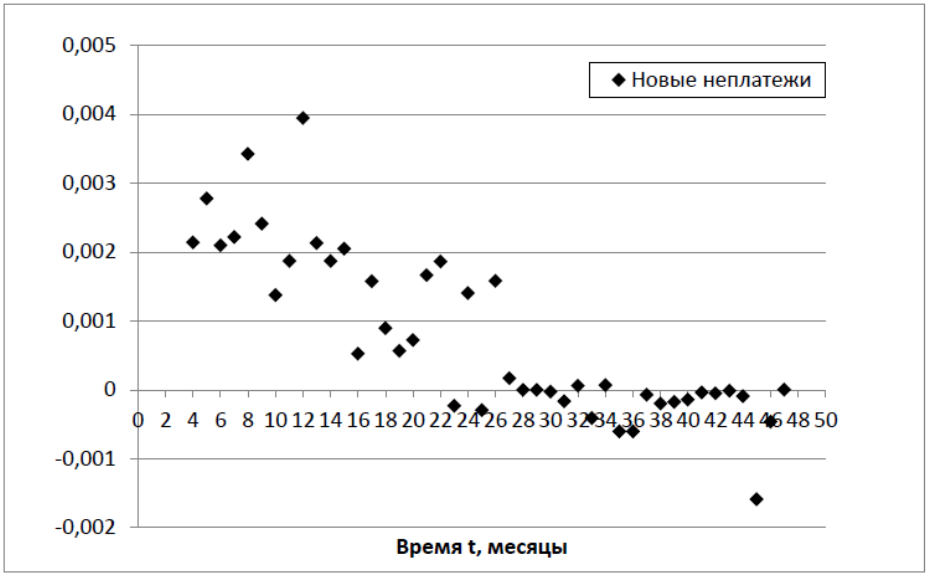

Рис. 1 – Доля новых невыплат по кредиту

Чтобы получить зависимость интенсивности отказов от времени, аппроксимируем имеющиеся данные некоторой функцией. Линейная функция достаточно точно аппроксимирует исходные данные с коэффициентом детерминации $R^2 = 0{,}7064$ (см. рис. 2).

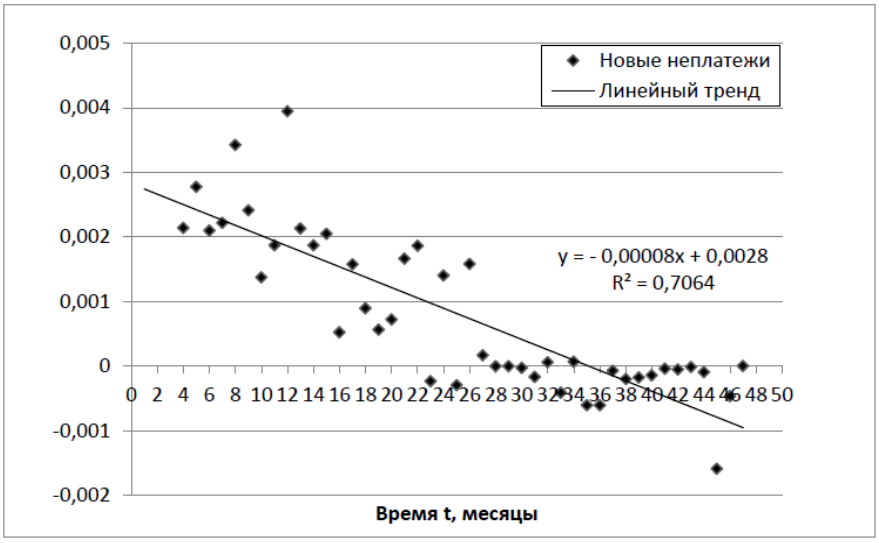

Рис. 2 – Аппроксимация доли новых невыплат по кредиту линейной функцией

При аппроксимации полиномом (см. рис. 3) с ростом степени улучшается точность приближения, коэффициент детерминации $R^2 = 0{,}7619$ (при приближении полином 6 степени). Однако впоследствии использование полиномиальной функции значительно усложняет вычисления.

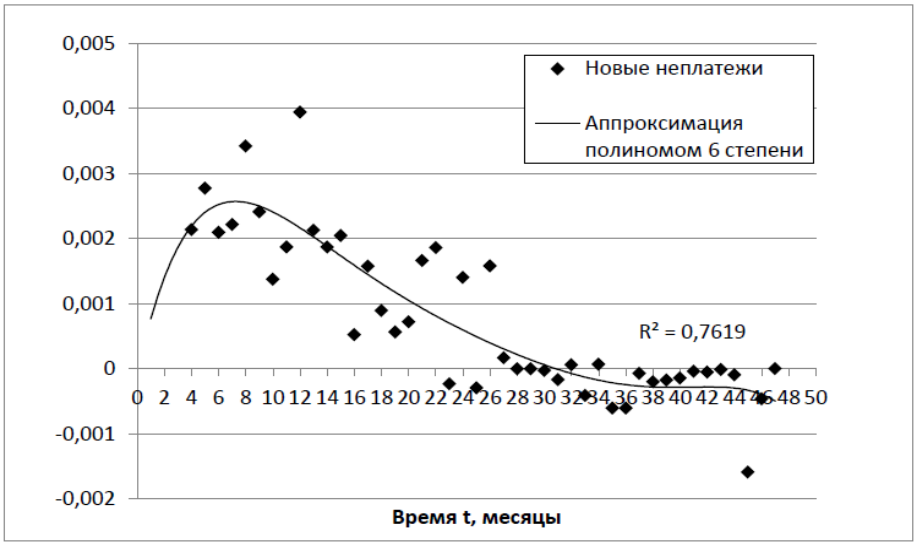

Рис. 3 – Аппроксимация доли новых невыплат по кредиту полиномиальной функцией

Будем предполагать, что интенсивность отказов для кредитных выплат описывается линейной функцией, которой аппроксимируются исходные данные по просроченным платежам

$$\mu(t) = -0{,}00008t + 0{,}0028.$$

Произведем ряд математических вычислений. Подставляя полученные данные в формулу для расчета актуарной приведенной стоимости страхования кредита, получаем

$$A_C^{(m)} = \frac{C}{i^{(m)}nm}\sum_{j=1}^{nm}\left[(1+i^{(m)})^{nm-j}-1\right]e^{(0{,}00004\left(\frac{j^2+1}{m^2}\right)+0{,}00008\left(\frac{j}{m^2}+\frac{x(j+1)}{m}\right)-0{,}0028\frac{(j+1)}{m})}.$$

Пусть сумма кредита $C = 100000$ руб., а годовая процентная ставка по нему $i = 15\%$, срок выплаты кредита – 4 года, выплаты производятся ежемесячно, $m = 12$. Подставляя эти данные, получаем, что стоимость страхования кредита составит 642 рубля 11 копеек. Стоимость полиса адекватна сумме кредита; исходя из полученных результатов, модель можно считать состоятельной.

4.2. Модификация модели в условиях кризисной ситуации

В ситуации повышенного риска неплатежеспособности заемщика модель может существенно измениться. Адаптируем модель к условиям кризисной ситуации. Будем применять аналитические методы, поскольку статистических данных в условиях кризиса недостаточно. Это происходит из-за краткосрочности кризисного периода в сравнении с продолжительностью стабильного развития экономики.

Во время кризиса ситуация в экономике негативно влияет на состояние финансов заемщиков, а соответственно и на повышение количества невыплат по кредитам. Поэтому функция интенсивности отказов также изменится, а ее скорость убывания уменьшится. Также, можно предположить, что свободный член функции увеличится, потому что количество неплатежей по кредитам существенно возрастет. Можно сделать вывод, что параметры функции интенсивности отказов изменятся (см. рис. 4).

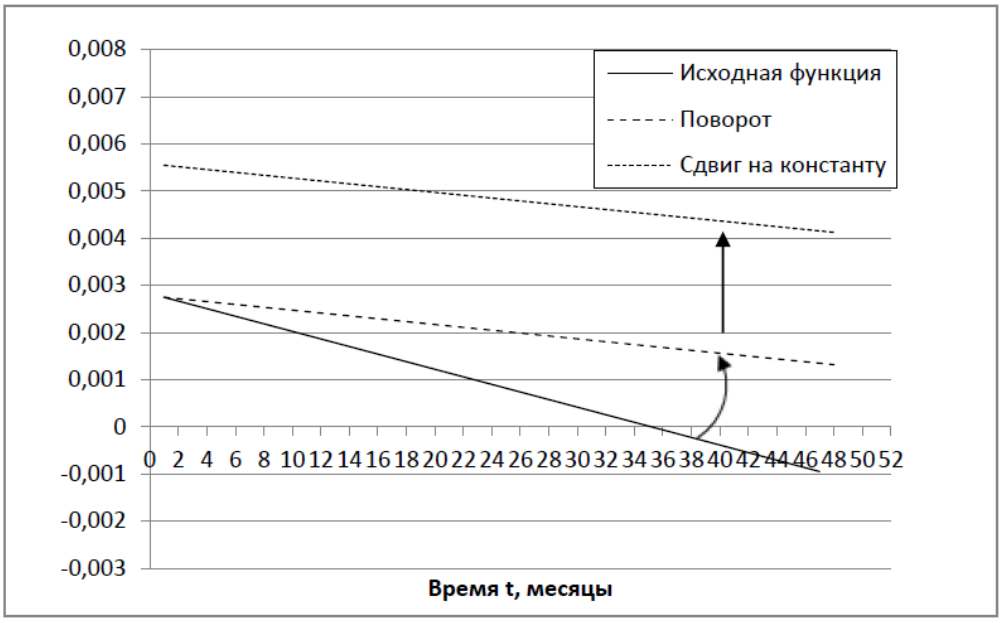

Рис. 4 – Сдвиг исходной функции на константу

Построив новую функцию, можно заметить, что она демонстрирует более негативный прогноз интенсивности отказов. Новая функция в общем виде выглядит следующим образом

$$\mu_{cr}(t) = \mu(t) + c(t - t_0) + d,$$

$\mu_{cr}(t) -$ интенсивность отказов во время кризиса, $\mu(t)$ – интенсивность отказов в стабильной экономической ситуации, $c(t) + d$ – линейная функция, t_0 – точка неподвижности во время поворота.

Так как уменьшается скорость убывания функции, то исходная функция совершает поворот вправо относительно неподвижной точки t_0. А также вследствие того, что доля неплатежей по кредитам резко увеличивается, происходит сдвиг на константу.

Пусть во время кризиса количество невыплат кредита возрастет в первоначальный момент в 1,5 раза, а скорость убывания уменьшится в 2 раза. Тогда новое выражение функции интенсивности отказов примет вид

$$\mu_{cr}(t) = -0{,}00004t + 0{,}0042.$$

Подставляя полученные данные в формулу для расчета актуарной приведенной стоимости страхования кредита, получаем

$$A_C^{(m)} = \frac{C}{i^{(m)}nm}\sum_{j=1}^{nm}\left[(1 + i^{(m)})^{nm-j} - 1\right] e^{\left(0{,}00002\left(\frac{j^2+1}{m^2}\right)+0{,}00004\left(\frac{j}{m^2}+\frac{x(j+1)}{m}\right)-0{,}0042\frac{(j+1)}{m}\right)}.$$

Оставим те же условия по кредиту: сумма кредита $C = 100000$ руб., годовая процентная ставка $i = 15\%$, срок выплаты кредита -4 года, выплаты производятся ежемесячно, $m = 12$. Подставив значения в формулу, получим, что стоимость страхования кредита равна 986 рублей 73 копейки. Таким образом, стоимость страхового полиса увеличилась в полтора раза.

§ 5. Оценка уровня неплатежеспособности заемщика

Потребительские виды кредита являются наиболее доходными видами кредитов, выдаваемыми банками. Но с другой стороны, именно они имеют очень большую степень кредитного риска, так как состояние финансов отдельного человека или семьи может резко ухудшиться вследствие утраты работы или наступления непредвиденных обстоятельств, требующих затрат денежных средств.

Снижение платежеспособности заемщиков приводит к уменьшению объемов банковских средств, поэтому необходимо более точно оценивать кредитные риски для расчета страховой премии. Можно оценить вероятность неплатежеспособности физического лица с помощью рейтинговой методики [5].

5.1. Описание модели

Этот способ позволяет определять вес каждого фактора в итоговой оценке в соответствии с его влиянием, выявленным статистическими методами. Его преимущество состоит и в том, что он позволяет учитывать одновременно множество факторов кредитоспособности. Для этого необходимо выставлять рейтинговый балл на этапе выдачи кредита и сопоставлять его с баллом рейтинговой группы, для которой заранее задана вероятность неплатежеспособности. Факторами будут выступать такие показатели как:

- Возраст
- Пол
- Образование

- Сумма кредита
- Срок кредита
- Уровень дохода
- Социальный статус
- Первый или повторный кредит
- Семейное положение
- Цель кредитования
- Сфера деятельности

Выбор факторов обусловлен тем, что их можно легко проверить по заполняемой анкете на выдачу кредита. Сбор информации по таким данным осуществляется постоянно, поэтому у страховых компаний и банков имеется большая база данных, которую они могут использовать для оценки. Такой большой набор факторов позволяет производить улучшенную классификацию, а соответственно формулировать выводы о каждой группе заемщиков.

Вероятность банкротства физического лица, обладающего определенными признаками, вычисляется так

$$P_B = \prod_i^n p_i, \quad (1)$$

где $p_i -$ вероятность банкротства заемщика, имеющего признак i; $n -$количество признаков. Важно заметить, что формула (1) верна только в случае слабой корреляции или ее полного отсутствия. Поэтому необходимо проверить признаки на корреляцию и исключить наименее значимый признак из рассмотрения.

Рейтинговый балл можно вычислить следующим образом

$$R = \sum_i^n r_i,$$

где $r_i -$ рейтинг заемщика, имеющего признак i.

Прологарифмируем обе части равенства (1), чтобы получить однозначную связь между рейтинговым баллом и вероятностью банкротства

$$log_a P_B = log_a(\prod_i^n p_i) = \sum_i^n log_a p_i.$$

Пусть $r_i = c * log_a p_i$, где $c -$ константа нормировки. Исходя из этого, рейтинговый балл

$$R = \sum_i^n r_i = c * \sum_i^n log_a p_i = c * log_a P_B. \quad (2)$$

Из формулы (2) можно выявить однозначную связь между рейтинговым баллом и вероятностью банкротства физического лица. Сопоставим рейтинг заемщика рейтинговой группе, которая характеризуется вероятностью банкротства. Таким образом, все физические лица разместятся по группам, которым однозначно сопоставится максимальная вероятность разорения.

Подробно рассмотрим способ выбора значимых некоррелированных признаков с помощью дерева решений, которое состоит из ребер и узлов. Дерево решений (см. Приложение 1) позволяет выделить наиболее значимые признаки с точки зрения увеличения количества информации. На ребрах дерева записаны признаки классификации (атрибуты), которые влияют на значения целевой функции, расположенные в листьях, а в других узлах также расположены атрибуты, по которым можно различить случаи классификации. Будем применять формулу для вычисления энтропии [8]. Рассмотрим множество A, состоящее из n объектов, где m из которых обладают свойством S, принимающее s разных значений. Таким образом, энтропия множества A по отношению к свойству S

$$H(A,S) = -\sum_{i=1}^{s} \frac{m_i}{n} * log_a \frac{m_i}{n}. \quad (3)$$

Атрибут будем выбирать таким образом, чтобы после классификации энтропия относительно целевой функции стала как можно меньше. Далее рассчитаем прирост информации. Пусть множество A элементов классифицируются с помощью атрибута Q, который имеет q разных значений. Следовательно, прирост информации

$$Gain(A,Q) = H(A,S) - \sum_{i=1}^{q} \frac{|A_i|}{|A|} * H(A_i,S), \quad (4)$$

где A_i – множество элементов из A, на которых имеет значение i атрибут Q. А количество информации, необходимое для разделения по текущему атрибуту

$$SplitInfo(A,Q) = -\sum_{i=1}^{q} \frac{|A_i|}{|A|} * log_a \frac{|A_i|}{|A|}. \quad (5)$$

Для выбора подходящего атрибута будем использовать следующий критерий, как максимизацию значения

$$GainRatio(A, Q) = \frac{Gain(A,Q)}{SplitInfo(A,Q)}. \quad (6)$$

Также, чтобы рассчитать прирост информации возможно использование индекса Гинни

$$Gini(A, S) = 1 - \sum_{i=1}^{S} \frac{|A_i|}{|A|}.$$

Поэтому для набора A, атрибута Q, имеющего q значений и целевого свойства S индекс вычисляется

$$Gini(A, Q, S) = Gini(A, S) - \sum_{j=1}^{q} \frac{|A_j|}{|A|} * Gini(A_j, S).$$

При построении дерева решений на каждом этапе происходит процедура расчета прироста информации, благодаря этому осуществляется упорядочивание признаков по влиянию на целевую функцию. В ходе расчета корреляции атрибутов легко исключаются коррелированные признаки исходя из наименьшего влияния на целевую функцию. Поэтому при расчете рейтинга участвуют только те некоррелированные или слабо коррелированные признаки, которые оказывают влияние на вероятность дефолта.

Деревья решений при накоплении статистики имеют возможность гораздо лучше классифицировать и выявлять скрытые связи между атрибутами. Для новой классификации нужно опуститься до листа и выдать соответствующее значение. Поэтому, они легко адаптируются к изменяющимся экономическим условиям.

После определения рейтинга заемщика и подбора для него подходящей рейтинговой группы необходимо определить параметры функции интенсивности для расчета страховой премии. Будем использовать функцию интенсивности Мейкхема, так как постоянное слагаемое A позволяет учесть риски для жизни, связанные с несчастными случаями, а член $Be^{\alpha x}$ учитывает влияние возраста на смертность. Заметим, что эта модель является нелинейной.

Будем проводить оценку параметров модели следующим образом [4]. Составим систему из трех уравнений с тремя неизвестными A, B и α, решив которую, мы получим оценки неизвестных параметров.

- Рассмотрим функцию распределения модели Мейкхема:

$$F(x) = 1 - s(x) = 1 - e^{-Ax-\frac{B(e^{\alpha x}-1)}{\alpha}};$$

- Найдем нижний x_{pn} и верхний x_{pv} квартиль и медиану x_{ps} распределения по нашим статистическим данным группы

$$P(X < x_p) = F(x_p) = 1 - S(x_p) = p; \quad (7)$$

- Тогда система нелинейных уравнений будет иметь вид

$$\begin{cases} 1 - e^{-Ax_{pn}-\frac{B\left(e^{\alpha x_{pn}}-1\right)}{\alpha}} = p_n \\ 1 - e^{-Ax_{pv}-\frac{B\left(e^{\alpha x_{pv}}-1\right)}{\alpha}} = p_v; \\ 1 - e^{-Ax_{ps}-\frac{B\left(e^{\alpha x_{ps}}-1\right)}{\alpha}} = p_s \end{cases} \quad (8)$$

- Решив систему, мы получим оценки параметров $\hat{A}$, $\hat{B}$ и $\hat{\alpha}$;
- Оцененная функция интенсивности будет иметь вид

$$\mu(x) = \hat{A} + \hat{B}e^{\hat{\alpha}x};$$

- Таким образом, получим формулу для стоимости страхования

$$A_C = \frac{C}{ni}\sum_{k=1}^{n}[(1+i)^{n-k}-1]\left[e^{-(\hat{A}k+\frac{\hat{B}e^{\hat{\alpha}x}\left(e^{\hat{\alpha}k}-1\right)}{\hat{\alpha}})} - e^{-(\hat{A}(k+1)+\frac{\hat{B}e^{\hat{\alpha}x}\left(e^{\hat{\alpha}(k+1)}-1\right)}{\hat{\alpha}})}\right]. \quad (9)$$

В случае, когда выплаты происходят непрерывно

$$A_C = \frac{C}{n(e^{\delta}-1)}\int_1^n \left[\left[(e^{\delta})^{n-t}-1\right]\left[e^{-(\hat{A}t+\frac{\hat{B}e^{\hat{\alpha}x}\left(e^{\hat{\alpha}t}-1\right)}{\hat{\alpha}})} - e^{-(\hat{A}(t+1)+\frac{\hat{B}e^{\hat{\alpha}x}\left(e^{\hat{\alpha}(t+1)}-1\right)}{\hat{\alpha}})}\right]\right]dt. \quad (10)$$

5.2. Алгоритм расчета страховой премии по кредиту на основании рейтинга заемщика

1. Первичный отбор

Изначально, заемщики заполняют анкету, на основании которой проводится скоринговая оценка, во время которой решается выдавать кредит

или нет. Также, в случае успешного прохождения отбора, назначается сумма кредита.

2. Расчет рейтинга

На втором шаге, клиентам, которые прошли первичный отбор проставляется рейтинг следующим образом.

2.1. На основании имеющейся статистики предыдущих выдач, временные рамки которых задаются аналитиком исходя из условий экономики, строится дерево решений, благодаря которому упорядочиваются признаки, оказывающие влияние на платежеспособность клиента.

2.1.1. Для этого производится расчет корреляции признаков, которые потенциально влияют на платежеспособность клиентов по формулам (3)–(6).

2.1.2. Коррелирующие факторы исключаются из рассмотрения на основании дерева решений, то есть, факторы, которые оказывают наименьшее влияние на возможность банкротства. При этом более значимые – остаются.

2.2. Далее рассчитывается рейтинг заемщика по формуле (2).

3. Определение рейтинговой группы

В зависимости от рассчитанного рейтинга клиент помещается в рейтинговую группу, которой однозначно сопоставляется вероятность разорения.

4. Расчет страховой премии

На этом этапе происходит определение вида функции интенсивности отказов, оценки параметров которой необходимы для расчета приведенной стоимости страхования кредита.

4.1. Идентифицируются оценки параметров функции интенсивности отказов.

4.1.1. Рассчитываются квартили распределения по формуле (7).

4.1.2. Решение системы уравнений (8) дает оценку параметров функции.

4.2. С помощью формулы (10) проводится расчет приведенной стоимости страхования кредита для определенной рейтинговой группы.

Таким образом, подсчет рейтинга заемщика на основании дерева решений позволяет гораздо более точно рассчитать стоимость страхового полиса. Также, этот алгоритм можно адаптировать и модифицировать под текущие экономические условия.

5.3. Примеры расчета

Рассмотрим реализацию программы на примере нескольких клиентов банка.

Пусть клиент берет в банке кредит на следующих условиях:

Срок кредита $n = 2$ года;

Процентная ставка $i = 5\%$ при дискретном начислении процентов;

Процентная ставка $\delta = 1\%$ при непрерывном начислении процентов;

Сумма кредита $C = 900000$ руб;

Личные параметры заемщика:

36	31	ж	высшее	900000	2	140000	работодате	повторный	женат	потреб	услуги

Тогда его рейтинг будет равен 56;

Рейтинговая группа [56, 56, 55, 56, 56, 56, 56, 56, 55, 55];

Актуарная приведенная стоимость страхования кредита при дискретном начислении процентов 1663.41823041972;

Актуарная приведенная стоимость страхования кредита при непрерывном начислении процентов 579.490870392493.

Второй заемщик имеет следующие характеристики:

Срок кредита $n = 3$ года;

Процентная ставка $i = 5\%$ при дискретном начислении процентов;

Процентная ставка $\delta = 1\%$ при непрерывном начислении процентов;

Сумма кредита $C =$100000 руб;

Личные параметры:

198	68	м	среднее	100000	3	40000	пенсионер	первый	женат	потреб	другое

Рейтинг кредитоспособности 36;

Рейтинговая группа [36, 43, 43];

Актуарная приведенная стоимость страхования кредита при дискретном начислении процентов 1457.83223151147;

Актуарная приведенная стоимость страхования кредита при непрерывном начислении процентов 1430.14765914857.

Если же процентная ставка по кредиту при дискретном начислении процентов $i = 10\%$, а процентная ставка $\delta = 2\%$ при непрерывном начислении процентов, то соответственно:

Актуарная приведенная стоимость страхования кредита при дискретном начислении процентов 1176.20445225382;

Актуарная приведенная стоимость страхования кредита при непрерывном начислении процентов 2177.93255278741.

Таким образом, можно сделать вывод, что страховая премия по полису адекватна сумме кредита. Нужно отметить, что при росте рейтинга актуарная приведенная стоимость страхования снижается. Процентная ставка также существенно влияет на стоимость полиса, при ее увеличении наблюдается та же тенденция относительно страховой премии.

Заключение

В ходе данной работы получены следующие результаты, которые выносятся на защиту:

1. Построена и изучена модель для нахождения актуарной приведенной стоимости страхования кредита в случае ежегодных и m-кратных выплат;

2. Проведена оценка построенной модели при помощи методов актуарной математики и данных банковской статистики в пакете Excel, вычислена актуарная приведенная стоимость страхования кредита;

3. Построена модифицированная модель в условиях кризисной ситуации и вычислена стоимость кредитного страхования;

4. Разработан алгоритм расчета рейтинга заемщика для расчета страховой премии на основании построения дерева решений;

5. Выполнена реализация калькулятора расчета страховой премии на языке Python.

Список литературы

1. Бауэрс Н., Гербер Х., Джонс Д., Несбитт С., Хикман Дж. Актуарная математика. Перевод с англ./Под ред. В.К. Малиновского. – М.: Изд-во. Янус-К, 2001. – 665 с.
2. Беляевских Е.А. Исследование и анализ кредитных рисков методами актуарной математики: дис. ... канд. экон. наук : – М.: МГУ им. М.В. Ломоносова, 2013. – 38 с.
3. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и ее инженерные приложения. – М.: Изд-во Наука, 1988. – 480 с.
4. Леонова О.В. Моделирование смертности населения с помощью аналитических законов на примере России. – Иркутск.: Известия Байкальского государственного университета. 2019. Т. 29, вып. 1, с. 95-106.
5. Петухова М.В. Рейтинговая методика оценки кредитного риска физических лиц. – Новосибирск.: Вестник НГУ. Серия: Социально – экономические науки. 2011. Т. 11, вып. 3, с.86-93.
6. Фалин Г.И., Фалин А.И. Введение в актуарную математику. – М.: Изд-во Финансово-актуарный центр МГУ им. М.В. Ломоносова, 1994. – 248 с.
7. Четыркин Е.М. Финансовая математика. – М.: Дело, 2004. – 400 с.
8. Шеннон К. Работы по теории информации и кибернетике. – М.: Изд-во ин. лит., 1963. – 832 с.
9. Andrew Kuritzkes, Til Schuermann, Scott Weiner. Deposit Insurance and Risk Management of the U.S. Banking System: How Much? How Safe? Who Pays? – The Wharton School, University of Pennsylvania, 2002. – 35 p.
10. Reza Vaez-Zadeh, Danyang Xie, Edda Zoli. A Market-Oriented Deposit Insurance Scheme. – International Monetary Fund, 2002 – 43 p.

11. Деревья решений: общие принципы [Электронный ресурс]: [2021]. – Режим доступа: https://loginom.ru/blog/decision-tree-p1, свободный. – Загл. с экрана. (01.05.2021).
12. Объединенное кредитное бюро [Электронный ресурс]: [2020]. – Режим доступа: https://bki-okb.ru, свободный. – Загл. с экрана. (01.06.2020).
13. C4.5 [Электронный ресурс]: [2021]. – Режим доступа: https://wiki2.org/ru/C4.5, свободный. – Загл. с экрана. (01.04.2021).
14. SciPy.org [Электронный ресурс]: [2021]. – Режим доступа: https://docs.scipy.org/doc/scipy/reference/index.html, свободный. – Загл. с экрана. (08.05.2021).
15. The Human Life-Table Database [Электронный ресурс]: [2020]. – Режим доступа: http://www.lifetable.de/, свободный. – Загл. с экрана. (07.06.2020).

Приложение 1

Теоретические аспекты алгоритма расчета страховой премии

Алгоритм расчета актуарной приведенной стоимости страхования кредита на основании вычисления рейтинга неплатежеспособности основан на построении дерева решений, которое базируется на классификаторе C4.5.

Деревья решений

Дерево решений [11] — это инструмент прогнозного моделирования, который можно применять во многих областях. Они являются наиболее мощными алгоритмами, которые подпадают под категорию контролируемых.

Обозначим набор некоторых понятий, используемых в дальнейшем.

Таблица 2 – Терминология дерева решений

Название	Описание
Объект	Пример, шаблон, наблюдение
Атрибут	Признак, независимая переменная, свойство
Целевая переменная	Зависимая переменная, метка класса
Узел	Внутренний узел дерева, узел проверки
Корневой узел	Начальный узел дерева решений
Лист	Конечный узел дерева, узел решений
Решающее правило	Условие в узле, проверка

Основная сущность дерева — это узлы принятия решений, где данные разделяются и удаляются, где мы получаем промежуточный результат.

Деревья решений, как средство классификации, используются уже давно. Развитие инструмента построения деревьев началось в 1950-х годах. Тогда были предложены основные идеи в области исследований моделирования человеческого поведения с помощью компьютерных систем.

Такой способ прогнозного моделирования имеет ряд преимуществ при использовании. Опишем некоторые достоинства деревьев решений:

1) Простота в восприятии

При построении дерева решений его результат достаточно легко может интерпретироваться пользователем. Дерево решений наглядно объясняет, почему и по каким критериям определенный объект был размещен в тот или иной класс.

2) Алгоритм сам выбирает атрибуты

Деревья решений не требуют выбора входных атрибутов. При построении используются все атрибуты, из которых алгоритм выбирает наиболее значимые и на их основе строит дерево решений.

3) Большой объем информации

Алгоритм дает возможность работать с большим объёмом информации без специальных процедур подготовки. Деревья решений не предполагают наличие специального оборудования для обработки больших баз данных.

4) Различные виды переменных

Деревья решений могут работать как с категориальными, так и с интервальными переменными.

Самым популярным методом построения деревьев решений является алгоритм C4.5, который будет далее использоваться при реализации. Остановимся на нем подробнее.

Алгоритм C4.5

Во время работы данного алгоритма [13] происходит разбиение области принимаемых значений независимой переменной на интервалы. Исходное множество разделяется на подмножества в соответствии с интервалом таким образом, чтобы значение зависимой переменной попало в данный промежуток.

Дерево решений строится сверху вниз, то есть, от корневого узла к листьям.

На первом шаге происходит обучение путем формирования «пустого» дерева, состоящего из одного корневого узла. Узел содержит все обучающее множество. Далее оно разбивается на подмножества, из которых позже

сформируются узлы-потомки посредством выбора одного из атрибутов и формированием правил.

В результате разбиения получаются некоторые подмножества и формируются потомки корневого узла, каждому из которых ставится в соответствие свое подмножество. Эта процедура применяется рекурсивно ко всем подмножествам до тех пор, пока не будет выполнено условие остановки обучения.

На втором шаге, после того, как на обучающем наборе данных построено дерево решений будем приступать к классификации новых объектов. Новый классифицируемый объект сначала попадает в корневой узел дерева, а далее перемещается по остальным узлам. В них проверяется соответствие между значением атрибута и правилом данного узла. После данных манипуляций исходный объект направляется в определенный узел-потомок. Процесс будет продолжаться до тех пор, пока классифицируемый объект не попадет в лист. В результате, ему присваивается метка класса, ассоциированная с данным листом.

Приложение 2

Реализация калькулятора расчета страховой премии

Данная программа реализует алгоритм расчета актуарной приведенной стоимости страхования кредита на основании вычисления рейтинга неплатежеспособности с помощью дерева решений на языке программирования Python. Дерево решений реализуется посредством алгоритма классификатора C4.5, описанного выше.

Обучающее дерево

Входные данные

На вход подаются обучающие данные о клиентах банка. Среди данных такие показатели, как: id (идентификатор личности), возраст, пол, образование, сумма кредита, срок кредита, уровень дохода, социальный статус, первый или

повторный кредит, семейное положение, цель кредитования, сфера деятельности, рейтинг (см. рис. 5).

id	возраст	пол	образова	сумма	срок	уровень д	соц статус	первый и	сем поло	цель	сфера дея	рейтинг
1	18	м	среднее	1000000	1	10000	студент	первый	не женат	потреб	другое	22
2	21	ж	среднее	300000	2	20000	студент	первый	женат	бизнес	другое	29
3	24	м	высшее	5000000	3	10000	безработ	повторны	не женат	бизнес	другое	34
4	27	м	высшее	50000	4	50000	по найму	первый	женат	потреб	образова	43
5	30	ж	среднее с	700000	5	600000	по найму	повторны	женат	потреб	финансы	52
6	33	м	среднее	800000	2	40000	по найму	первый	женат	потреб	гос.управ	36
7	36	ж	среднее	1000000	3	200000	работода	повторны	не женат	потреб	it	46
8	39	м	среднее	300000	4	20000	безработ	повторны	не женат	потреб	другое	33
9	42	ж	высшее	2000000	5	80000	работода	повторны	женат	потреб	финансы	55
10	45	м	среднее с	3000000	4	70000	по найму	первый	женат	бизнес	образова	38
11	48	ж	высшее	600000	1	30000	работода	повторны	не женат	потреб	услуги	45
12	51	м	среднее с	30000	2	100000	по найму	первый	женат	потреб	услуги	35
13	54	ж	среднее	400000	3	20000	безработ	повторны	не женат	потреб	другое	35
14	57	м	высшее	1500000	4	20000	пенсионе	первый	женат	потреб	другое	36
15	60	ж	среднее	300000	1	15000	пенсионе	повторны	женат	потреб	другое	36
16	63	м	среднее с	90000	2	30000	по найму	повторны	не женат	потреб	услуги	35
17	66	ж	высшее	400000	0,5	30000	по найму	повторны	женат	бизнес	образова	46
18	69	м	среднее	1000000	0,3	100000	работода	первый	женат	потреб	финансы	36
19	72	ж	среднее	100000	5	40000	пенсионе	повторны	женат	потреб	другое	41
20	75	м	среднее с	90000	1	10000	пенсионе	повторны	женат	потреб	другое	34
21	78	ж	среднее	5000000	0,6	15000	пенсионе	повторны	не женат	потреб	другое	33
22	81	м	среднее	400000	2	20000	пенсионе	первый	не женат	потреб	другое	25
23	84	ж	среднее	20000	1	30000	пенсионе	повторны	не женат	бизнес	другое	33
24	87	м	среднее	200000	3	20000	пенсионе	первый	женат	бизнес	другое	26
25	18	м	среднее с	700000	0,7	15000	студент	первый	не женат	потреб	другое	24
26	19	ж	среднее	1000000	2	20000	студент	первый	женат	потреб	другое	32
27	22	ж	высшее	300000	3	60000	по найму	повторны	не женат	потреб	it	43
28	25	м	высшее	1000000	5	100000	по найму	повторны	не женат	потреб	гос.управ	46

Рис. 5 – Обучающие данные

Выходные данные

По имеющимся данным алгоритм обучается классифицировать клиентов. На выходе получим следующий результат, который представляет собой дерево решений в текстовом варианте. Фигурными скобками обозначены его ветви, а атрибуты заключены в одинарные кавычки. Если после атрибута расположены двоеточие и фигурные скобки, значит, идет ветвление дерева. В обратном случае, если после атрибута стоит число, следовательно, получаем рейтинг клиента.

{'цель': {'бизнес': {'сем положение': {'не женат': {'сфера деятельности': {'услуги': **39**, 'другое': {'пол': {'м': {'образование': {'высшее': {'возраст': {'от 50': **27**, 'от 23 до 40': **34**}}, 'среднее': **18**}}, 'ж': **33**}}}}, 'женат': {'пол': {'м': {'первый или повт': {'первый': {'образование': {'среднее специальное': **38**, 'среднее': {'возраст': {'от 50': **26**, 'от 23 до 40': **38**}}}}, 'повторный': **41**}}, 'ж': {'первый или повт': {'первый': **29**, 'повторный': {'образование': {'высшее': {'возраст': {'от 50': **46**, 'от 40 до 50': **51**}}, 'среднее специальное': **51**}}}}}}}}, 'потреб': {'первый или повт': {'первый': {'пол': {'м': {'сем положение': {'не женат': {'сфера деятельности': {'финансы':

38, 'другое': {'возраст': {'до 22': {'образование': {'среднее специальное': **24**, 'среднее': **22**}}, 'от 50': **25**}}}}, 'женат': {'образование': {'высшее': {'возраст': {'от 50': **36**, 'от 23 до 40': **43**}}, 'среднее специальное': **35**, 'среднее': {'сумма': {'от 300 до 1 млн': **36**, 'от 100 до 300 тыс': {'возраст': {'от 23 до 40': **35**, 'от 40 до 50': **29**}}}}}}}}, 'ж': {'возраст': {'до 22': **32**, 'от 50': **43**}}}}, 'повторный': {'пол': {'м': {'сем положение': {'не женат': {'соц статус': {'по найму': {'образование': {'высшее': {'возраст': {'от 23 до 40': **46**, 'от 40 до 50': **42**}}, 'среднее специальное': **35**}}, 'безработный': {'возраст': {'от 50': **27**, 'от 23 до 40': **33**}}}}, 'женат': {'образование': {'высшее': **44**, 'среднее специальное': {'сумма': {'от 100 до 300 тыс': **40**, 'до 100 тыс': **34**}}}}}}}, 'ж': {'сем положение': {'не женат': {'образование': {'высшее': {'возраст': {'до 22': **43**, 'от 40 до 50'**: 45**}}, 'среднее': {'возраст': {'до 22': **46**, 'от 50': {'уровень дохода': {'от 15 до 30': {'сумма': {'от 300 до 1млн': **35**, 'до 100 тыс': **36**}}, 'до 15 тыс': **33**}}}}}}}, 'женат': {'соц статус': {'работодатель': {'возраст': {'от 23 до 40': **56**, 'от 40 до 50': **55**}}, 'по найму': {'сумма': {'от 300 до 1млн': {'сфера деятельности': {'образование': **46**, 'финансы': {'возраст': {'от 50': **33**, 'от 23 до 40': {'образование': {'высшее': {'срок': {'4': **53**, '2': **50**}}, 'среднее специальное': **52**}}}}}}, 'от 100 до 300 тыс': **43**, 'до 100 тыс': {'возраст': {'от 23 до 40': **47**, 'от 40 до 50': **43**}}}}, 'пенсионер': {'срок': {'4': **41**, '2': **36**}}}}}}}}}}}

Код

```
class DecisionTree:
    def loadData(self, path):
        dataSet = pd.read_csv(path, delimiter=',')
        labelSet = list(dataSet.columns.values)
        dataSet = dataSet.values
        return dataSet, labelSet

    def calcShannonEnt(self, dataSet):
        numEntries = len(dataSet)
        labelCounts = {}
        for featVec in dataSet:
            currentLabel = featVec[-1]
            if currentLabel not in labelCounts.keys():
                labelCounts[currentLabel] = 1
            labelCounts[currentLabel] += 1

        shannonEnt = 0.0
        for key in labelCounts:
```

```
        prob = float(labelCounts[key]) / numEntries
        shannonEnt -= prob * np.log2(prob)
    return shannonEnt

def splitDataSet(self, dataSet, axis, value):
    retDataSet = []
    for featVec in dataSet:
        if featVec[axis] == value:
            reduceFeatVec = list(featVec[:axis])
            reduceFeatVec.extend(featVec[axis + 1:])
            retDataSet.append(reduceFeatVec)
    return retDataSet

def chooseBestFeatureID3(self, dataSet):
    numFeature = len(dataSet[0]) - 1
    baseEntropy = self.calcShannonEnt(dataSet)
    bestInfoGain = 0.0
    bestFeature = -1
    for i in range(numFeature):
        featureList = [example[i] for example in dataSet]
        uniqueVals = set(featureList)
        newEntropy = 0.0
        for value in uniqueVals:
            subDataSet = self.splitDataSet(dataSet, i, value)
            prob = len(subDataSet) / float(len(dataSet))
            newEntropy += prob * np.log2(prob)
        infoGain = baseEntropy - newEntropy

        if infoGain > bestInfoGain:
            bestInfoGain = infoGain
            bestFeature = i
    return bestFeature

def chooseBestFeatureC45(self, dataSet):
    numFeature = len(dataSet[0]) - 1
    baseEntropy = self.calcShannonEnt(dataSet)
    bestInfoGainRatio = 0.0
    bestFeature = -1
```

```
    for i in range(numFeature):
        featureList = [example[i] for example in dataSet]
        uniqueVals = set(featureList)
        newEntropy = 0.0
        for value in uniqueVals:
            subDataSet = self.splitDataSet(dataSet, i, value)
            prob = len(subDataSet) / float(len(dataSet))
            newEntropy += prob * np.log2(prob)
        infoGain = baseEntropy - newEntropy

        splitInfo = 0.0
        for value in uniqueVals:
            subDataSet = self.splitDataSet(dataSet, i, value)
            prob = len(subDataSet) / float(len(dataSet))
            splitInfo += prob * np.log2(prob)

        infoGainRatio = infoGain / (-splitInfo)
        if infoGainRatio > bestInfoGainRatio:
            bestInfoGainRatio = infoGainRatio
            bestFeature = i
    return bestFeature

def majorityCnt(self, classList):
    classCount = {}
    for vote in classList:
        if vote not in classCount.keys():
            classCount[vote] = 0
        classCount += 1

    sortedClassCount = sorted(classCount.items(), key=operator.itemgetter(1), reverse=True)
    return sortedClassCount[0][0]

def createTree(self, dataSet, labels, method):
    classList = [example[-1] for example in dataSet]
    if classList.count(classList[0]) == len(classList):
        return classList[0]
         if len(dataSet[0]) == 1:
        return self.majorityCnt(classList)
```

```
        if method == "ID3":
            bestFeat = self.chooseBestFeatureID3(dataSet)
        elif method == "C4.5":
            bestFeat = self.chooseBestFeatureC45(dataSet)
        else:
            bestFeat = self.chooseBestFeatureID3(dataSet)

        bestFeatLabel = labels[bestFeat]
        myTree = {bestFeatLabel: {}}
        del (labels[bestFeat])
        featValues = [example[bestFeat] for example in dataSet]
        uniqueValues = set(featValues)
        for value in uniqueValues:
            subLabels = labels[:]
            myTree[bestFeatLabel][value] = self.createTree(self.splitDataSet(dataSet, bestFeat, value),
subLabels, method)
        return myTree
```

Классификатор

После обучения программа умеет классифицировать и выставлять рейтинг новым клиентам. Также программа определяет заемщиков в рейтинговую группу, которой однозначно сопоставляется максимальная вероятность разорения.

Входные данные

На вход программе подаются те же данные о клиентах, что и в обучающей части, за исключением информации о рейтинге (см. рис. 6).

id	возраст	пол	образование	сумма	срок	уровень дс	соц статус	первый или п	сем положе	цель	сфера деятельност
1	18	ж	среднее спец	90000	5	30000	безработн	повторный	не женат	потреб	другое
2	29	м	среднее	5000000	2	100000	студент	первый	женат	бизнес	другое
3	31	ж	высшее	400000	3	20000	по найму	повторный	не женат	бизнес	образование
4	33	м	среднее	20000	4	20000	по найму	первый	женат	потреб	образование
5	30	ж	высшее	200000	5	20000	по найму	повторный	женат	потреб	финансы
6	33	м	среднее	700000	4	10000	по найму	первый	женат	потреб	гос.управление
7	36	ж	среднее спец	1000000	1	50000	работодат	повторный	не женат	потреб	it
8	39	м	высшее	5000000	2	600000	безработн	повторный	не женат	бизнес	другое
9	42	ж	среднее	400000	3	40000	работодат	первый	женат	бизнес	финансы
10	45	м	среднее	20000	4	70000	по найму	повторный	женат	потреб	образование
11	48	ж	высшее	200000	1	30000	работодат	повторный	не женат	потреб	услуги
12	51	м	высшее	700000	2	100000	работодат	повторный	женат	потреб	услуги
13	54	ж	среднее спец	1000000	0,5	20000	безработн	первый	не женат	потреб	другое
14	29	м	среднее	300000	0,3	20000	по найму	повторный	женат	потреб	услуги
15	31	ж	среднее	1000000	5	20000	по найму	первый	женат	потреб	образование
16	33	м	среднее спец	30000	1	10000	по найму	повторный	не женат	бизнес	финансы
17	66	ж	высшее	5000000	0,5	50000	пенсионер	повторный	женат	потреб	другое
18	69	м	высшее	60000	0,3	600000	пенсионер	первый	женат	потреб	другое
19	72	ж	среднее спец	100000	5	40000	пенсионер	повторный	женат	потреб	другое
20	75	м	среднее	90000	1	140000	пенсионер	повторный	женат	потреб	другое
21	78	м	среднее спец	5000000	0,6	60000	пенсионер	повторный	не женат	потреб	другое
22	81	ж	среднее спец	400000	5	120000	пенсионер	повторный	не женат	потреб	другое
23	84	м	высшее	20000	4	10000	пенсионер	первый	не женат	бизнес	другое
24	87	м	высшее	200000	3	10000	пенсионер	повторный	женат	потреб	другое
25	75	ж	высшее	700000	0,8	100000	пенсионер	первый	не женат	потреб	другое
26	19	м	среднее спец	1000000	2	20000	студент	повторный	женат	потреб	другое
27	22	м	среднее	300000	0,6	20000	по найму	повторный	не женат	потреб	it
28	25	ж	высшее	1000000	5	15000	по найму	повторный	не женат	потреб	гос.управление

Рис. 6 –Тестовые данные

Выходные данные

На выходе получим информацию о рейтинге определенного клиента и баллах рейтинговой группы, в которую он попал.

Rating

Client ID: 18

Rating: 36

Period: 2

Amount: 60000

Group Members Ratings: [43, 43, 36]

Client ID: 131

Rating: 56

Period: 2

Amount: 7000000

Group Members Ratings: [56, 56, 56, 55, 56, 56, 56, 56, 55, 55]

Код

```
class Classificator:
    def __init__(self):
        self.dataSet = []
        self.labelSet = []
        self.tree = None
        self.finalData = dict()
        self.groupsData = None
        self.test_calc = dict()

    def loadData(self, path, groups_path, tree):
        self.dataSet = pd.read_csv(path, delimiter=',')
        self.labelSet = list(self.dataSet.columns.values)
        self.dataSet = self.dataSet.values
        self.tree = tree

        with open(groups_path, 'r', encoding='utf-8') as json_file:
            self.groupsData = json.load(json_file)

    def classificate(self):
        first_key = list(self.tree.keys())[0]
        index_id = 1
        for row_data in self.dataSet:
            try:
                self.proccessBranch(first_key, self.tree, row_data, index_id, list())
            except KeyError:
                self.finalData[index_id] = None
            finally:
                index_id += 1

        self.groupRatedData(1)
        return self.finalData

    def defineGroup(self, name, data):
        groups_data = self.groupsData[name]
```

```
    for group in groups_data.keys():
        if (groups_data[group][0] < data) and (groups_data[group][1] > data):
            return group
    return None

def proccessBranch(self, key, tree_data, test_row, index_id, guid):
    guid.append(key)
    key_value = ''
    groups_list = ['срок', 'возраст', 'сумма', 'уровень дохода']
    client_data = dict()
    client_data['Period'] = 0
    client_data['Amount'] = 0
    for i in range(0, len(self.labelSet)):
        if self.labelSet[i] == key:
            if key in groups_list:
                key_value = self.defineGroup(key, test_row[i])
            else:
                key_value = test_row[i]
            break
    guid.append(key_value)

    for i in range(0, len(self.labelSet)):
        if self.labelSet[i] == 'срок':
            client_data['Period'] = test_row[i]
        if self.labelSet[i] == 'сумма':
            client_data['Amount'] = test_row[i]

    if isinstance(tree_data[key][key_value], dict):
        new_key = list(tree_data[key][key_value].keys())[0]
        self.proccessBranch(new_key, tree_data[key][key_value], test_row, index_id, guid)
    else:
        client_data['Rating'] = tree_data[key][key_value]
        client_data['GUID'] = guid
        client_data['GroupMembers'] = []
        client_data['Chances'] = []
```

```
        client_data['Quartiles'] = []
        client_data['CalcRating'] = []
        client_data['Factors'] = dict()
        client_data['Integral'] = ()
        client_data['Insurance (9)'] = 0.0
        client_data['Insurance (10)'] = 0.0
        self.finalData[index_id] = client_data
        del guid

def groupRatedData(self, depth):
    for item in self.finalData:
        last_group_num = 0
        if self.finalData[item] is not None:
            last_group_num = len(self.finalData[item]['GUID']) - (1 + depth)
        if last_group_num > 0:
            for iter_item in self.finalData:
                eq_flag = True
                if (self.finalData[iter_item] is not None) and (item != iter_item):
                    for i in range(0, last_group_num):
                        try:
                            self.finalData[iter_item]['GUID'][i]
                        except IndexError:
                            eq_flag = False
                            break
                        else:
                        self.finalData[iter_item]['GUID'][i])
                            if self.finalData[item]['GUID'][i] != self.finalData[iter_item]['GUID'][i]:
                                eq_flag = False
                                break
                    if eq_flag:
                        self.finalData[item]['GroupMembers'].append(iter_item)
```

Идентификатор оценок параметров

После того как информация о клиенте оказалась в определенной рейтинговой группе, происходит идентификация оценок параметров с помощью расчета вероятностей разорения, квартилей распределения и решения системы нелинейных уравнений с помощью численного метода библиотеки Python.

Входные данные

Идентификатор получает на вход данные о рейтингах клиентов определенной группы.

$Group\ Members\ Ratings$

Выходные данные

После ряда процедур программа определяет параметры системы, необходимые для дальнейших расчетов страховой премии.

$Chances$

$Quartiles$

$CalcRating$

$Factors$

Client ID: 18

Rating: 36

Period: 2

Amount: 60000

Group Members Ratings: [43, 43, 36]

Chances: [0.013568559012200934, 0.013568559012200934, 0.02732372244729256, 0.02732372244729256]

Quartiles: [0.013568559012200934, 0.020446140729746747, 0.02732372244729256]

CalcRating: [43.0, 38.899611316744874, 36.0]

Factors: {'A': 0.01740439580720837, 'B': 2.370139465834168e-07, 'alfa': -0.8176710477040533}

Client ID: 131

Rating: 56

Period: 2

Amount: 7000000

Group Members Ratings: [56, 56, 56, 55, 56, 56, 56, 56, 55, 55]

Chances: [0.003697863716482932, 0.003697863716482932, 0.003697863716482932, 0.003697863716482932, 0.003697863716482932, 0.003697863716482932, 0.003697863716482932, 0.003697863716482932, 0.004086771438464067, 0.004086771438464067, 0.004086771438464067]

Quartiles: [0.003697863716482932, 0.003697863716482932, 0.0038923175774734993]

CalcRating: [56.0, 56.0, 55.48750520486374]

Factors: {'A': 0.0030139211688776407, 'B': 6.717059196182931e-05, 'alfa': 0.1226239762014517}

Код

```
def equations(vars, data):
    a, b, alfa = vars

    eq1 = 1 - exp(-a * data[0][0] - ((b * (exp(data[0][0] * alfa) - 1)) / alfa)) - 0.25
    eq2 = 1 - exp(-a * data[0][1] - ((b * (exp(data[0][1] * alfa) - 1)) / alfa)) - 0.50
    eq3 = 1 - exp(-a * data[0][2] - ((b * (exp(data[0][2] * alfa) - 1)) / alfa)) - 0.75
    return [eq1, eq2, eq3]



def integrand(t, A, B, alfa, delta, n , x):
    return (exp(delta) ** (n - t) - 1) * (exp(-(A * t + (B * exp(alfa * x) * (exp(alfa * t) - 1)) / alfa)) - exp(-(A *
(t + 1) + (B * exp(alfa * x) * (exp(alfa * (t + 1)) - 1)) / alfa)))
if __name__ == '__main__':
    decTree = DecisionTree()
    cls = Classificator()
    dataSet, labelSet = decTree.loadData('D:\\Work\\PythonProjects\\riski\\data\\1data_ru_grp_new.cvs')
    tree = decTree.createTree(dataSet, labelSet, "C4.5")
    print('Полученное дерево решений:')
    print(tree)

    cls.loadData('D:\\Work\\PythonProjects\\riski\\data\\1data_test_ru_new.cvs',
                 'D:\\Work\\PythonProjects\\riski\\data\\groups.txt',
                 tree)
```

```
ratedData = cls.classificate()

for item in ratedData:
    pi_list = list()
    if ratedData[item] is not None:
        pi = float(exp(ratedData[item]['Rating'] / -10))
        pi_list.append(pi)
        for cl_id in ratedData[item]['GroupMembers']:
            next_pi = float(exp(ratedData[cl_id]['Rating'] / -10))
            pi_list.append(next_pi)
            pi_list.sort()
            ratedData[item]['Chances'] = pi_list

        cld = {'rate': ratedData[item]['Chances']}
        df = pd.DataFrame(data=cld)
        qp = df.rate.quantile([0.25, 0.5, 0.75])
        ratedData[item]['Quartiles'].append(qp[0.25])
        ratedData[item]['Quartiles'].append(qp[0.50])
        ratedData[item]['Quartiles'].append(qp[0.75])

        P25 = float(-10 * np.log(qp[0.25]))
        P50 = float(-10 * np.log(qp[0.50]))
        P75 = float(-10 * np.log(qp[0.75]))

        data = (P25, P50, P75)
        ratedData[item]['CalcRating'].append(P25)
        ratedData[item]['CalcRating'].append(P50)
        ratedData[item]['CalcRating'].append(P75)

        alfa_waitings = 0.1
        a, b, alfa = None, None, None

        for i in range(1, 10):
            a, b, alfa = fsolve(equations, (0, 0, alfa_waitings), args=[data])
            if alfa != alfa_waitings:
                if (a <= 0) or (b <= 0) or (alfa <= 0):
                    alfa_waitings = alfa_waitings + 0.1
                else:
```

```
            break
        else:
            alfa_waitings = alfa_waitings + 0.1

    ratedData[item]['Factors']['A'] = a
    ratedData[item]['Factors']['B'] = b
    ratedData[item]['Factors']['alfa'] = alfa
```

Расчет премии

Входные данные

Для того чтобы произвести расчет актуарной приведенной стоимости страхования кредита с дискретным и непрерывным начислением процентов, на вход программе подаются данные о параметрах функции интенсивности и параметрах кредита.

$a, b, alfa, n, \delta, C, i, x$

Выходные данные

В результате работы программы получим два вида актуарной приведенной стоимости страхования кредита, соответствующие формулам алгоритма (9) и (10).

$Insurance9$

$Insurance10$

Client ID: 18 Rating: 36 Period: 2 Amount: 60000 Insurance 9: 610.422708572596 Insurance 10: 211.522168618056
Client ID: 131 Rating: 56 Period: 2 Amount: 7000000 Insurance 9: 12937.6973477089 Insurance 10: 4507.15121416384

Код

```
for item in ratedData:
    if ratedData[item] is not None:
        A = ratedData[item]['Factors']['A']
        B = ratedData[item]['Factors']['B']
        alfa = ratedData[item]['Factors']['alfa']
        delta = 2
        n = int(ratedData[item]['Period'])
        x = 0
        C = ratedData[item]['Amount']
        i = 10

        ratedData[item]['Integral'] = integrate.quad(integrand, 1, n, args=(A, B, alfa, delta, n, x))
        ratedData[item]['Insurance (10)'] = C / (n * (exp(delta) - 1)) * ratedData[item]['Integral'][0]

        summ = 0
        for k in range(1, n):
            summ += ((1 + i) ^ (n - k) - 1) * (exp(-(A * k + (B * exp(alfa * x) * (exp(alfa * k) - 1) / alfa))) - exp(-(A * (k + 1) + (B * exp(alfa * x) * (exp(alfa * (k + 1)) - 1) / alfa))))
        ratedData[item]['Insurance (9)'] = (C / (n * i)) * summ # * (exp(-(A * k + (B * exp(alfa * x) * (exp(alfa * k) - 1) / alfa))) - exp(-(A * (k + 1) + (B * exp(alfa * x) * (exp(alfa * (k + 1)) - 1) / alfa))))

    for item in ratedData:
        if ratedData[item] is not None:
            print('Client ID: ' + str(item))
            print('Rating: ' + str(ratedData[item]['Rating']))
            print('Period: ' + str(ratedData[item]['Period']))
            print('Amount: ' + str(ratedData[item]['Amount']))
            gr_ratings = []
            for i in ratedData[item]['GroupMembers']:
                gr_ratings.append(ratedData[i]['Rating'])
            print('Group Members Ratings: ' + str(gr_ratings))
            print('Chances: ' + str(ratedData[item]['Chances']))
            print('Quartiles: ' + str(ratedData[item]['Quartiles']))
```

```
print('CalcRating: ' + str(ratedData[item]['CalcRating']))
print('Factors: ' + str(ratedData[item]['Factors']))
print('Integral: ' + str(ratedData[item]['Integral']))
print('Insurance 9: ' + str(ratedData[item]['Insurance (9)']))
print('Insurance 10: ' + str(ratedData[item]['Insurance (10)']))
print('-----------------------------------------------')
```