

Федеральное государственное бюджетное учреждение высшего образования  
«Санкт-Петербургский государственный университет»

Факультет прикладной математики – процессов управления

**Итоговый проект по дисциплине «Неклассические  
логики»:**

**«Разработка кредитного калькулятора с  
использованием нечёткой логики»**

Выполнил:

Березиков Алексей

Группа: 22.Б11-пу

Санкт-Петербург

2025

# Содержание

<b>1</b>	<b>Введение</b>	<b>2</b>
<b>2</b>	<b>Постановка задачи</b>	<b>2</b>
<b>3</b>	<b>Математическое обоснование</b>	<b>3</b>
<b>4</b>	<b>Описание и фаззификация входных переменных</b>	<b>3</b>
4.1	Входные переменные	3
4.2	Выходные переменные	4
<b>5</b>	<b>Построение базы нечётких лингвистических правил</b>	<b>5</b>
<b>6</b>	<b>Описание системы</b>	<b>6</b>
6.1	Входные и выходные переменные	6
6.2	Нечёткая логика	6
6.3	Расчёт рекомендуемой суммы	6
6.4	Графики	7
<b>7</b>	<b>Реализация</b>	<b>7</b>
7.1	Технологии	7
7.2	Структура проекта	8
7.3	Интерфейс	8
7.4	Доступность и адаптивность	8
<b>8</b>	<b>Анализ системы</b>	<b>8</b>
<b>9</b>	<b>Графики зависимостей</b>	<b>9</b>
<b>10</b>	<b>Результаты</b>	<b>10</b>
10.1	Функциональность	10
10.2	Пример использования	10
<b>11</b>	<b>Заключение</b>	<b>11</b>

## Аннотация

Данный отчёт посвящён разработке веб-приложения для оценки вероятности одобрения кредита и рекомендуемой суммы на основе нечёткой логики. Пользователь вводит данные (доход, долговая нагрузка, кредитная история, возраст) через интуитивный интерфейс с иконками, всплывающими подсказками и слайдерами. Форма ввода фиксируется при прокрутке на десктопе. Система вычисляет результаты, отображает их с помощью прогресс-баров и визуализирует зависимости через интерактивные графики, включая зависимости от всех входных переменных и функции принадлежности. Проект реализован с использованием React, TypeScript, Material-UI и Redux. В отчёте описаны теоретические основы, постановка задачи, математическое обоснование, реализация, анализ и перспективы.

## 1 Введение

В условиях роста кредитных продуктов банки нуждаются в автоматизированных системах оценки кредитоспособности. Нечёткая логика, предложенная Лотфи Заде в 1965 году [1], позволяет моделировать сложные зависимости с учётом неопределённости, что делает её подходящей для кредитного скоринга.

Цель проекта — разработать веб-приложение, использующее нечёткую логику для оценки вероятности одобрения кредита и рекомендуемой суммы. Приложение предоставляет адаптивный интерфейс с фиксированной формой, иконками, анимациями и графиками. В отчёте рассматриваются теоретические основы, постановка задачи, реализация, анализ и перспективы.

## 2 Постановка задачи

Задача проекта:

- Принимает входные данные:
  - Доход (0–200,000 руб.);
  - Долговая нагрузка (0–50,000 руб.);
  - Кредитная история (плохая, средняя, хорошая);
  - Возраст (18–70 лет).
- Вычисляет выходные параметры:
  - Вероятность одобрения (0–100%);
  - Рекомендуемая сумма (руб.).
- Визуализирует зависимости и функции принадлежности.

Система реализована как веб-приложение с адаптивным интерфейсом, где форма вво-

да фиксируется на десктопе. Вычисления основаны на нечёткой логике (фаззификация, правила, дефаззификация).

### 3 Математическое обоснование

Нечёткое множество — это множество пар  $\{x, \mu_A(x)\}$ , где  $\mu_A(x) \in [0, 1]$  — степень принадлежности. Трапецеидальная функция принадлежности:

$$\mu_A(x) = \begin{cases} 0, & x < a \text{ или } x > d \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \end{cases}$$

Алгоритм Мамдани:

1. **Фаззификация:** Преобразование входов в степени принадлежности.
2. **Правила:** Лингвистические правила «Если-то».
3. **Агрегация:** Минимум для конъюнкции, максимум для дизъюнкции.
4. **Дефаззификация:** Метод центроида:

$$y = \frac{\int \mu(y) \cdot y \, dy}{\int \mu(y) \, dy}$$

## 4 Описание и фаззификация входных переменных

### 4.1 Входные переменные

1. **Доход** (рис. 1):
  - Диапазон:  $[0; 200,000]$  руб.
  - Терм-множество: Очень низкий, Низкий, Средний, Высокий.
  - Функция принадлежности для «Средний»:

$$\mu_{\text{Средний}}(x) = \max \left( 0, \min \left( 1, \frac{x - 50000}{30000}, \frac{120000 - x}{70000} \right) \right)$$

2. **Долговая нагрузка** (рис. 2):
  - Диапазон:  $[0; 50,000]$  руб.
  - Терм-множество: Очень низкая, Низкая, Средняя, Высокая.
3. **Кредитная история:**

- Категории: Плохая, Средняя, Хорошая.
- Функции принадлежности: Дискретные (0 или 1).

#### 4. Возраст (рис. 3):

- Диапазон: [18; 70] лет.
- Терм-множество: Очень низкий, Низкий, Средний, Высокий, Очень высокий.

## 4.2 Выходные переменные

#### 1. Вероятность одобрения (рис. 4):

- Диапазон: [0; 100] %.
- Терм-множество: Низкая, Средняя, Высокая.
- Функция принадлежности для «Средняя»:

$$\mu_{\text{Средняя}}(x) = \max \left( 0, \min \left( 1, \frac{x - 20}{30}, \frac{80 - x}{30} \right) \right)$$

#### 2. Рекомендуемая сумма: Зависит от дохода, вероятности, долговой нагрузки, возраста и кредитной истории.

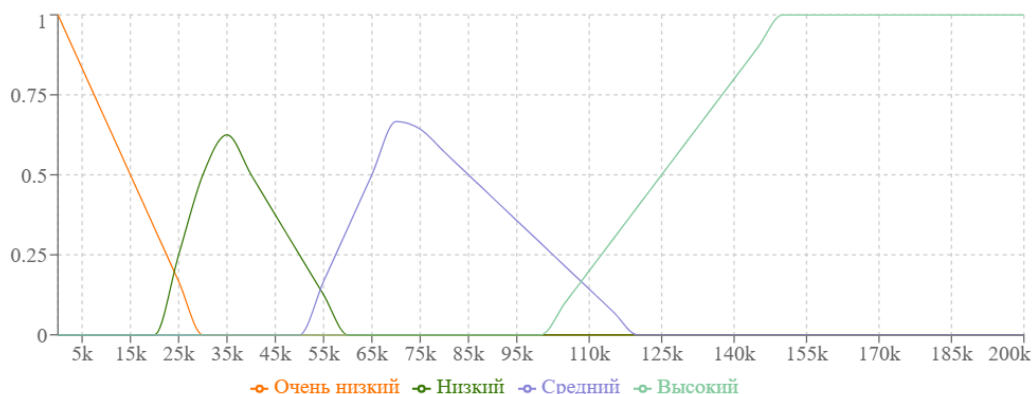


Рис. 1: Функции принадлежности для дохода

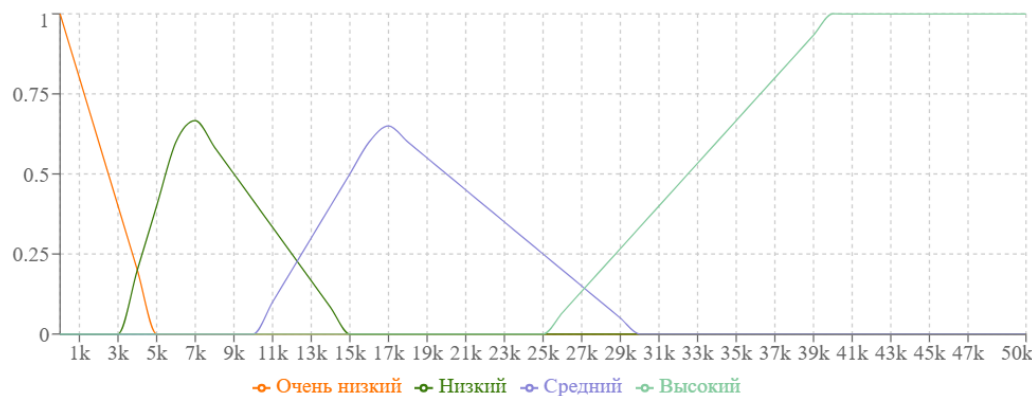


Рис. 2: Функции принадлежности для долговой нагрузки

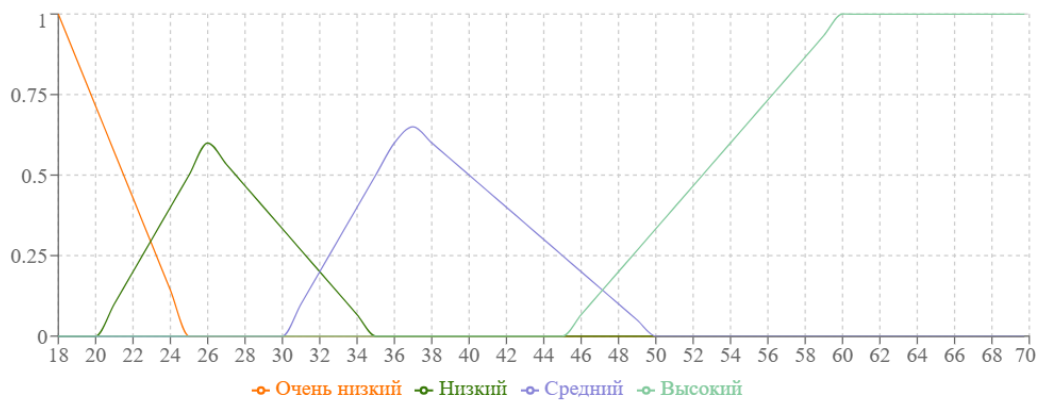


Рис. 3: Функции принадлежности для возраста

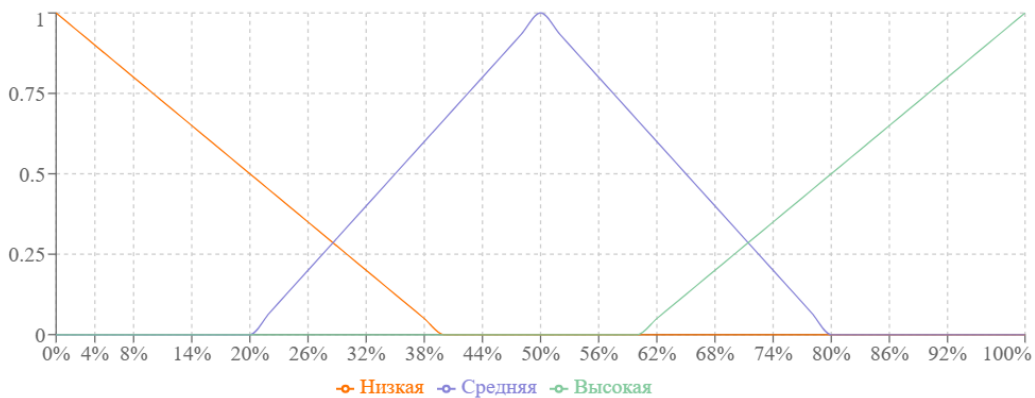


Рис. 4: Функции принадлежности для вероятности

## 5 Построение базы нечётких лингвистических правил

База знаний включает 192 правила, сгенерированных автоматически в файле `fuzzyRules.ts`. Правила формируются следующим образом:

### 1. Определение термов:

- Доход: Очень низкий, Низкий, Средний, Высокий (4 терма).
- Долговая нагрузка: Очень низкая, Низкая, Средняя, Высокая (4 терма).
- Кредитная история: Плохая, Средняя, Хорошая (3 терма).
- Возраст: Очень низкий, Низкий, Средний, Высокий (4 терма).
- Выход (вероятность): Низкая, Средняя, Высокая (3 терма).

### 2. Генерация комбинаций: Все возможные комбинации термов ( $4 \times 4 \times 3 \times 4 = 192$ ) создаются путём перебора.

### 3. Присвоение баллов:

- Доход: Очень низкий = 0, Низкий = 1, Средний = 2, Высокий = 3.
- Долг (обратная шкала): Очень низкая = 3, Низкая = 2, Средняя = 1, Высокая = 0.

- Кредитная история: Плохая = 0, Средняя = 1, Хорошая = 2.
  - Возраст: Очень низкий = 0, Низкий = 1, Средний = 2, Высокий = 1.
4. **Сумма баллов:** Суммируются баллы всех входных термов (максимум 10).
5. **Определение выхода и веса:**
- Если сумма  $\geq 7$ : Выход = Высокая, вес =  $0.9 - (8 - \text{сумма}) \cdot 0.05$ .
  - Если  $4 \leq \text{сумма} < 7$ : Выход = Средняя, вес =  $0.6 - (6 - \text{сумма}) \cdot 0.05$ .
  - Если сумма  $< 4$ : Выход = Низкая, вес =  $0.3 - (4 - \text{сумма}) \cdot 0.05$ .
6. **Применение:** Правила используют алгоритм Мамдани. Сила правила — минимум степеней принадлежности входов. Дефаззификация выполняется через взвешенную сумму центров тяжести термов (20%, 50%, 90% для Низкая, Средняя, Высокая).

Пример правил:

Доход	Долг	История	Возраст	Вероятность
Высокий	Низкая	Хорошая	Средний	Высокая
Низкий	Высокая	Плохая	Низкий	Низкая
Средний	Средняя	Средняя	Средний	Средняя
Очень низкий	Высокая	Плохая	Очень низкий	Низкая

Таблица 1: Пример нечётких лингвистических правил

## 6 Описание системы

### 6.1 Входные и выходные переменные

См. раздел 4.

### 6.2 Нечёткая логика

См. раздел 3.

### 6.3 Расчёт рекомендуемой суммы

Рекомендуемая сумма кредита вычисляется с учётом всех входных переменных:

$$\text{Сумма} = \text{Доход} \cdot \frac{\text{Вероятность}}{100} \cdot k \cdot f_{\text{долг}} \cdot f_{\text{возраст}} \cdot f_{\text{история}}$$

Где:

- $k$  — коэффициент масштабирования: 6 (если вероятность  $\geq 80\%$ ), 5 (если  $\geq 50\%$ ), иначе 3.
- $f_{\text{долг}} = 1 - \frac{\text{Долг}}{\text{Доход}}$  — снижение при высокой долговой нагрузке.

- $f_{\text{возраст}} = 0.8$  (если возраст  $< 30$ ),  $0.7$  (если возраст  $> 60$ ), иначе  $1$ .
- $f_{\text{история}} = 1$  (хорошая),  $0.8$  (средняя),  $0.5$  (плохая).

Сумма округляется до 10,000 руб. и ограничивается диапазоном 0–1,000,000 руб.

Пример: Доход = 100,000 руб., Долг = 10,000 руб., История = Хорошая, Возраст = 35, Вероятность = 90%.

- $f_{\text{долг}} = 1 - \frac{10,000}{100,000} = 0.9$ .
- $f_{\text{возраст}} = 1$  (возраст 35).
- $f_{\text{история}} = 1$  (хорошая).
- $k = 6$  (вероятность 90%).
- Сумма =  $100,000 \cdot \frac{90}{100} \cdot 6 \cdot 0.9 \cdot 1 \cdot 1 = 486,000 \approx 490,000$  ру.

## 6.4 Графики

Графики реализованы с помощью Recharts (рис. 5):

- Зависимость вероятности от дохода (рис. 6).
- Зависимость вероятности от долговой нагрузки (рис. 7).
- Зависимость вероятности от возраста (рис. 8).
- Зависимость вероятности от кредитной истории (рис. 9).
- Функции принадлежности для дохода, долга, возраста и вероятности (рис. 1–4).

График функций принадлежности для кредитной истории не включён, так как она является категориальной переменной с дискретными значениями (Плохая, Средняя, Хорошая), и её функции принадлежности (0 или 1) лучше представлены через график зависимости вероятности от кредитной истории (рис. 9).

## 7 Реализация

### 7.1 Технологии

- **React** [2]: Интерфейс.
- **TypeScript**: Типизация.
- **Material-UI** [3]: Дизайн.
- **Redux Toolkit**: Состояние.
- **Recharts**: Графики.
- **Jest**: Тестирование.



## 7.2 Структура проекта

- components/: CreditForm.tsx, CreditResult.tsx.
- fuzzyLogic/: fuzzySystem.ts, fuzzyRules.ts.
- store/: creditSlice.ts.
- theme.ts: Тема Material-UI.

## 7.3 Интерфейс

Интерфейс включает (рис. 5):

- **Форма ввода (CreditForm.tsx):** Поля с иконками (MonetizationOn, History, и т.д.), всплывающими подсказками и слайдерами. Фиксируется на десктопе (position: sticky).
- **Результаты (CreditResult.tsx):** Вероятность (прогресс-бар), сумма, вклад переменных (прогресс-бары), графики.

Раскладка: столбец на мобильных, строка (1:2) на десктопе.

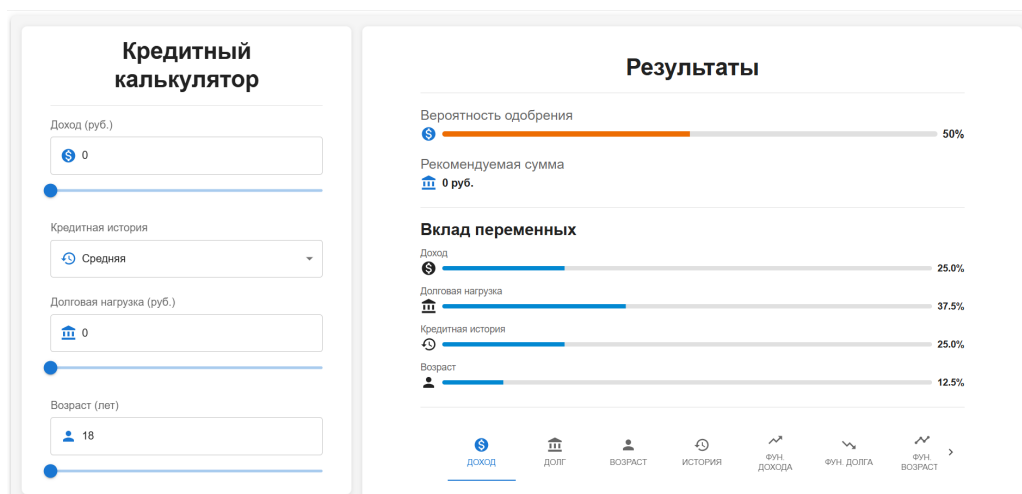


Рис. 5: Интерфейс приложения

## 7.4 Доступность и адаптивность

- ARIA-атрибуты.
- Breakpoints (xs, md).
- Полноширинные поля и прокручиваемые вкладки графиков.

## 8 Анализ системы

Результаты сравнивались с банковскими критериями:

№	Доход (руб.)	Долг (руб.)	История	Возраст	Модель (%)	Ожидание (%)
1	100,000	10,000	Хорошая	35	90	80–90
2	50,000	40,000	Плохая	25	20	15–25
3	150,000	5,000	Средняя	45	90	85–95
4	30,000	20,000	Хорошая	60	50	50–65

Таблица 2: Сравнение результатов модели и ожиданий

## 9 Графики зависимостей

- **Доход** (рис. 6): Вероятность растёт до 90% при 150,000 руб.

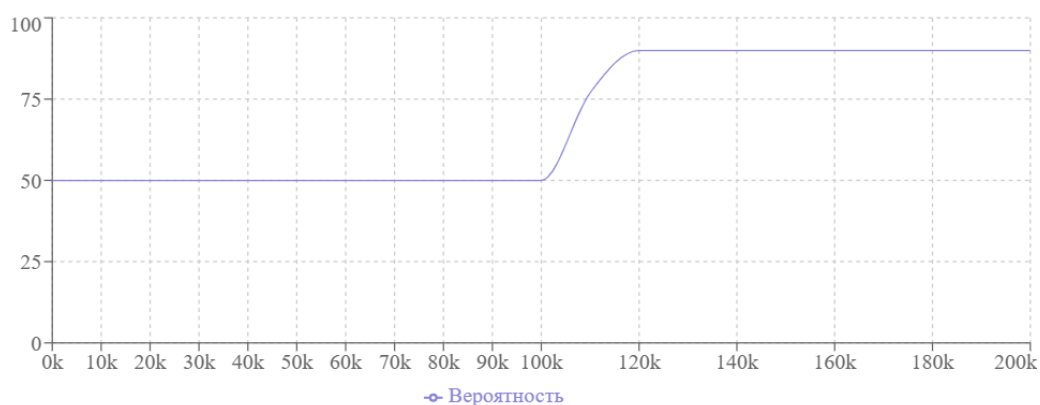


Рис. 6: Зависимость вероятности одобрения от дохода

- **Долговая нагрузка** (рис. 7): Падение вероятности при долге >30,000 руб.

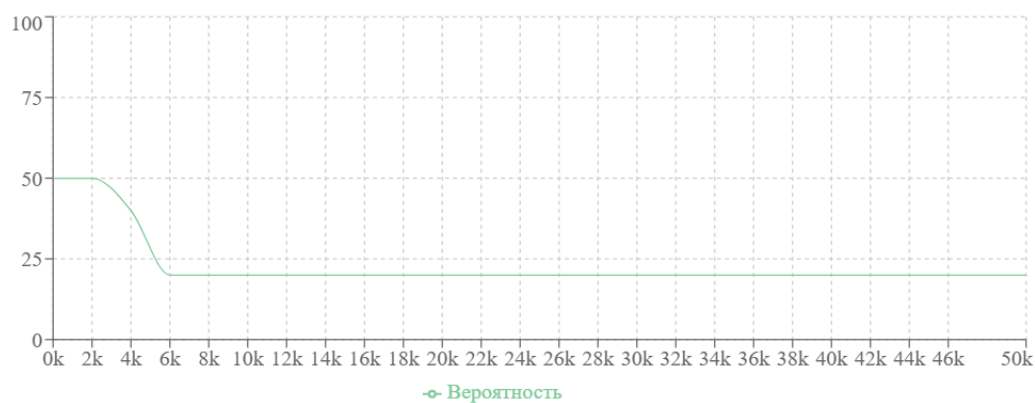


Рис. 7: Зависимость вероятности одобрения от долговой нагрузки

- **Возраст** (рис. 8): Пик вероятности в 30–50 лет.
- **Кредитная история** (рис. 9): Вероятность растёт от «Плохая» к «Хорошая».

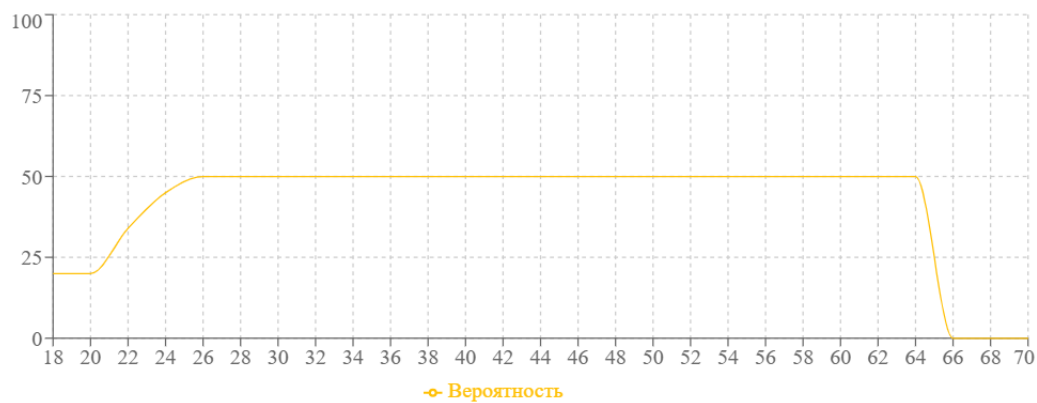


Рис. 8: Зависимость вероятности одобрения от возраста

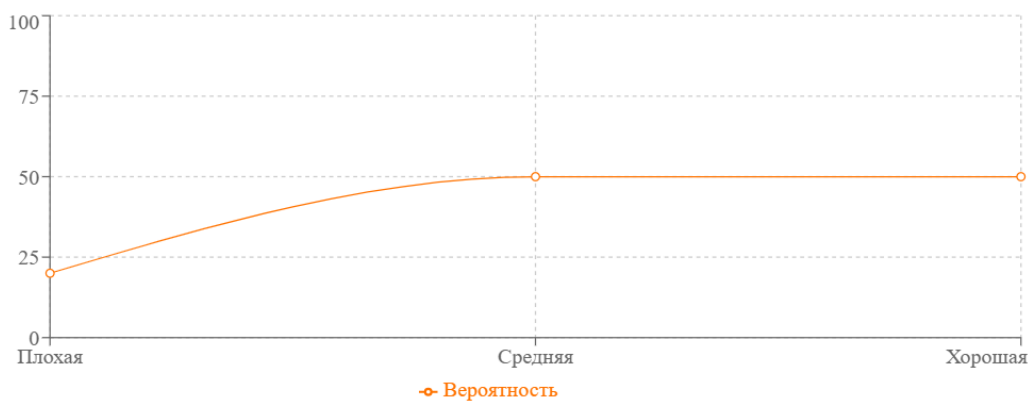


Рис. 9: Зависимость вероятности одобрения от кредитной истории

## 10 Результаты

### 10.1 Функциональность

Приложение позволяет вводить данные, вычислять результаты и визуализировать зависимости. Новая формула расчёта рекомендуемой суммы учитывает доход, долговую нагрузку, возраст и кредитную историю, что делает оценку более точной и соответствующей банковским практикам. Анимации, иконки и прокручиваемые вкладки улучшают пользовательский опыт.

### 10.2 Пример использования

Входные данные:

- Доход: 100,000 руб.
- Долговая нагрузка: 10,000 руб.
- Кредитная история: Хорошая.
- Возраст: 35 лет.

Результат (рис. 10):

- Вероятность: 90%.
- Сумма: 490,000 руб.
- Вклад: Доход (19,1%), Долг (13,9%), История (50,2%), Возраст (16,7%).

## Результаты

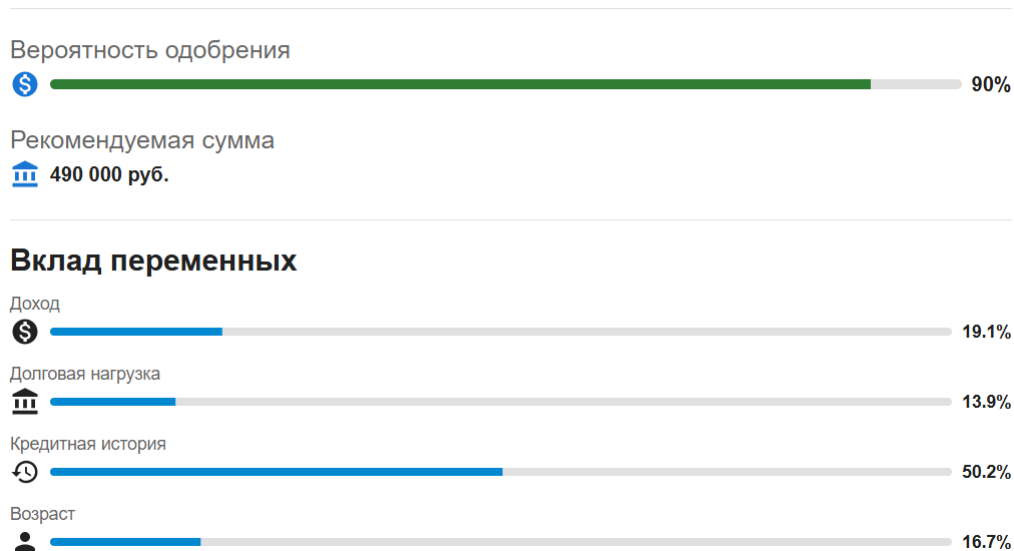


Рис. 10: Результаты вычислений

## 11 Заключение

Приложение успешно решает задачу кредитного скоринга с использованием нечёткой логики. Достижения:

- Точная модель с унифицированными термами (3 терма для вероятности).
- Улучшенный расчёт рекомендуемой суммы, учитывающий все входные переменные.
- Адаптивный интерфейс с фиксированной формой, иконками и анимациями.
- Полный набор графиков, включая зависимости от кредитной истории.

## Список литературы

- [1] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*.
- [2] React Documentation. <https://react.dev/>
- [3] Material-UI Documentation. <https://mui.com/>