Практика

проект – приложени за предсказване на това дали ще бъде одобрена молба за заем

Иван Я. Стойчев

2025

Съдържание

[1. За приложението 2](#_Toc202296561)

[2. За данните 2](#_Toc202296562)

[3. За модела 2](#_Toc202296563)

[4. Интеграция с уеб портала 2](#_Toc202296564)

[5. User stories 3](#_Toc202296565)

[ Администратор 3](#_Toc202296566)

[ Клиент 3](#_Toc202296567)

[ Потребител 3](#_Toc202296568)

[6. Sprint log 4](#_Toc202296569)

[7. Диаграма на базата данни 5](#_Toc202296570)

# За приложението

* Приложението представлява уеб портал, разработен с FastAPI за бекенд, React за фронтенд и PostgreSQL за база данни. Бекендът е изграден по многослойна архитектура, като се придържа към REST конвенциите. Уеб сайтът комуникира със сървъра чрез заявки, изпращани с помощта на Axios. За дизайна е използвана библиотеката shadcn — колекция от готови за интегриране компоненти. Уеб интерфейсът предоставя форма, на базата на която се извършва предсказване според въведените данни.

# За данните

* Данните за обучение на модела са взети от [Kaggle](https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset)
* Данните се състоят от общо 13 характеристики, като 1 от тях е целевата – ‚Approved‘ или ‚ Rejected ‘
* Преди тренирани на модела данните трябваше да бъдат приведени в технически коректен вид:
  + Нямаше липсващи наблюдения
  + Целевата характеристика (loan\_status) беше премахната
  + Качествените характеристика вяха превърнати в числови такива
  + След това данните бяха нормализирани

# За модела

* Моделът, който беше използван е бинарна логистична регресия
* Моделът постигна точност от 94% и с прецизност от 96%
* Моделът е трениран върху 1000 епохи, с learning rate от 0.0001

# Интеграция с уеб портала

* След като потребител се регистрира, може да попълни формата, след което информацията бива изпратена в JSON формат към бакенда, след което се предава на модел, който на база на тази информация извършва предсказани и връща на потребителя дали молбата му е по-вероятно да бъде одобрена или отказана

# User stories

## Администратор

* + Като администратор бих искал да мога да изтривам потребители
  + Като администратор бих искал да мога да променим лични данни на друг потребител
  + Като администратор бих искал да мога да вдигам ролята на нормален потребител до администратор
  + Като администратор бих искал да мога преглеждам предсказанията на други потребители
  + Като администратор бих искал да мога да изтривам предсказанията на други потребители

## Клиент

* + Като клиент би иска да мога лесно да попълня формата
  + Като клиент бих иска да мога да оставя обратна връзка в случай да проблем

## Потребител

* + - Като потребител бих искал да мога да се регистрирам
    - Като потребител бих искал да мога да сменя името си
    - Като потребител бих искал да мога да използвам уеб портала лесно от телефон

# Sprint log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Спринт | Период | Цели | Изпълнено | Затруднения |
| 1 | 3-9 юни | * Избор на тема * Намиране на дата сет * Структуриране на FastAPI проекта * Създаване на репо чрез github classroom | * Избран проект: Loan Approval Predictor * Избран дата сет: Взет от Kaggle * Създаване на базов FastAPI с докер и локална база данни | Нямаше затруднения |
| 2 | 10-16 юни | * Имплементация на логистична регресия с NumPy * Обучение и валидация * Интерфейс за вход | * Моделът е имплементиран без sklearn * Проведени тестове с различни параметри | Оптимизиране на модела |
| 3 | 17–23 юни | * Регистрация и логин * Потребителски роли * Интеграция на модела * Анкета и запазване на данни | * Роли: потребител и администратор * AI модул прави предсказания по вход от формата * Резултатите се съхраняват в базата | Нямаше затруднения |
| 4 | 24–30 юни | * Иплементиране на фронтенд страници * Финализиране на документацията * Github отчетност * Deploy-ване на бакенд и фронтенд | * Финализирани: README, диаграма * Deploy-ване на проекта на Azure * Deploy-ване на базата данни * Въвеждана на mock up информация в базата | Затруднение с качването на бакендa в azure |

# Картина, която съдържа текст, екранна снимка, диаграма, номер Генерираното от ИИ съдържание може да е неправилно.Диаграма на базата данни