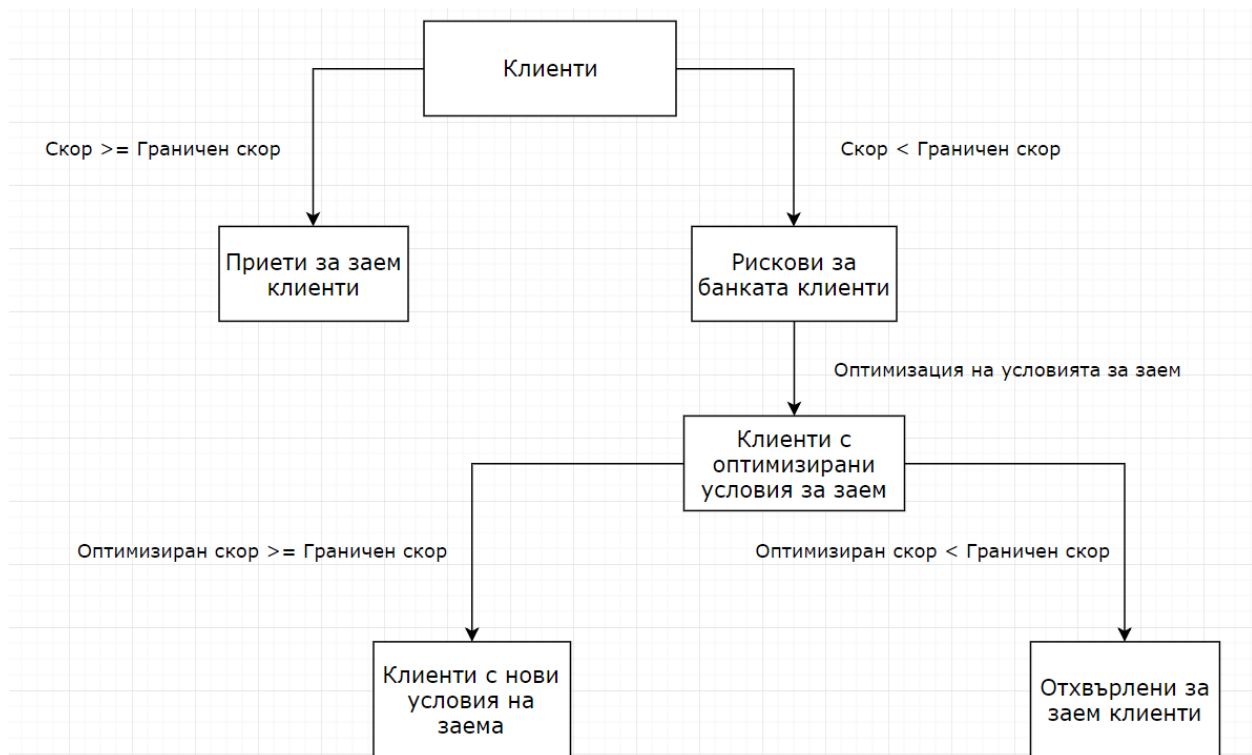


Разработка на Python

и пример

Проектът е съставен от файл с данни на клиенти (data.csv), код който се изпълнява (project.py) и ръчно разработена библиотека (project_library.py). Тази диаграма показва работния процес на програмата:



Проектът работи с над 18 хил. записа, но за целите на документа ще проследим пътят на три от тези записа:

ClientId	LoanAmount0	LoanPeriod0	Good	PredictedGood0
100001	2100	7	1	365
100003	7000	16	0	260
115932	5000	13	0	-441

Етапи на работа на програмата

1) Прочитане на данните и запазването им в подходящи структури от данни

Трите клиенти са прочетени от .csv файл и запазени в структура от данни наречена “data frame” идваща от най-популярната библиотека на езикът Python – “pandas”.

```
import pandas

import numpy as np

import project_library as pl

rootFilePath =
'C:/AleksandarUser/Programming/GitHubRepositories/TU-Data-
Mining/Project/Data/'

dataFilePath = rootFilePath + 'data.csv'

df = pandas.read_csv(dataFilePath)
```

“df” – това ще е променливата, която държи всички прочетени от файла данни за клиентите.

2) Филтриране на клиентите в две групи: първоначално приети клиенти и първоначално отхвърлени такива.

```
acceptedClientsDf = pl.get_accepted_clients(df, cutOff)

riskyClientsDf = pl.get_risky_clients(df, cutOff)
```

В променливата “acceptedClientsDf” стоят всички клиенти, които са първоначално приети. Такъв ще е клиент с идентификационен номер 100001. В “riskyClientsDf” пък ще се пазят рисковите клиенти, между които ще са клиентите с и.н. 100003 и 101164.

3) Оптимизиране на условията за заем

Ползвайки библиотеката за изчисления „GEKKO“, която извършва упоменатите във втори документ оптимизиращи изчисления, се изчисляват новите параметри за кредит на рисковите клиенти. Новите условия на двата рискови клиенти, които проследяваме, придобиват такъв вид:

ClientId	Rounded Optimized Loan Amount	Rounded Optimized Loan Period	Optimized Client Score With Rounded Values
100003	6900	18	380.65
115932	0	0	0

Първият рисков клиент получава нови закръглени стойности за кредит, които са по-малко рискови за предприятието.

Вторият клиент обаче получава нулеви стойности на условията. Това е индикатор, че алгоритъмът не е успял да направи нужните оптимизации, защото този клиент е твърде рисков.

Оптимизациите върху над 7 000 изискват по-различен подход. Тъй като на “Gekko” библиотеката ѝ трябва по 6 секунди на една оптимизация (близо 12 часа за всички записи), е вероятно да се получи грешка повреме на изчисленията и да се наложи да се започне наново. Затова по-удачно е да се правят изчисленията на части.

В долния код се определят имената на колоните във файлът и после се извършват по сто изчисления и записвания на резултатите във файл.

```
optimizedClientsFilePath = rootFilePath +  
'optimized_clients_data.csv'  
  
pl.define_csv_file_columns([  
    'ClientId',  
    'LoanAmount0',  
    'LoanPeriod0',  
    'Good',  
    'PredictedGood0',  
    'OptimizedLoanAmount',  
    'OptimizedLoanPeriod',  
    'OptimizedPredictedGood'  
], optimizedClientsFilePath)  
  
pl.optimize_and_write_to_csv_in_chunks(riskyClientsDf, par,  
cutOff, 100, optimizedClientsFilePath)
```

4) Филтриране на клиентите с оптимизирани условия на заем

След тежките изчисления, всички клиенти с оптимизирани условия на заема са в един файл. Данните се прочитат в променлива.

```
optimizedClientsDf = pandas.read_csv(optimizedClientsFilePath)
```

В нова променлива се разпределят клиентите, които след изчислен клиентски скор, на базата на новите условия на заем, преминават прагът от 350. Такъв ще е клиентът с и.н. 100003.

```
acceptedClientsAfterOptimizationDf =  
pl.get_accepted_clients_after_optimization(optimizedClientsDf,  
cutOff)
```

Тези клиенти биват запазени в нова променлива, която съдържа същите данни, но в закръглен вид.

```
roundedAttributesAcceptedClients =  
pl.replace_optimized_attributes_with_rounded_values(acceptedClientsAfterOptimizationDf, par)
```

Клиентите, чиито условия на заем не са били успешно оптимизирани се палят в друга променлива. Клиентът с и.н. 115932 е между тях.

```
deniedClientsAfterOptimizationDf =  
pl.get_denied_clients_after_optimization(optimizedClientsDf, cutOff)
```

5) Запазване резултатите

Първо се дефинират пътищата на файловете, където ще се запазят данните

```
acceptedClientsFilePath = rootFilePath + 'accepted_clients.csv'  
  
acceptedClientsAfterAdjustmentsFilePath = rootFilePath +  
'accepted_clients_after_optimizations.csv'  
  
deniedClientsAfterAdjustmentsFilePath = rootFilePath +  
'denied_clients_after_optimizations.csv'
```

После променливите, съдържащи резултатите се записват

```
acceptedClientsDf.to_csv(acceptedClientsFilePath, index = False)  
  
roundedAttributesAcceptedClients.to_csv(acceptedClientsAfterAdjustmentsFilePath, index = False)  
  
deniedClientsAfterOptimizationDf.to_csv(deniedClientsAfterAdjustmentsFilePath, index = False)
```

По този начин трите клиенти от началото се намират в три различни файлове. Първоначално приетият клиент с и.н. 100001 се пази във файл 'accepted_clients.csv'. Клиентът с и.н. 100003, който е успешно оптимизиран се пази в 'accepted_clients_after_optimizations.csv', а клиентът с и.н. 115932, който не беше оптимизиран в 'denied_clients_after_optimizations.csv'.

Анализи и заключение

Финалният файл, който е включен в проекта 'analysis.csv' показва анализ на данните на потенциално изпуснатите клиенти и по-точно - разликата между исканият от тях кредит и предложеният. По-важните колони са:

- dLAmount - процентно намаление на големината на заема
- dLPeriod - процентно увеличение на периода на заема
- Reject2Асепт - флаг: 0 - няма да се възползва от предложението, 1 - ще се възползва от предложението

Пресметнати са в проценти разликите между оптимизираните условия за кредит и първоначално исканите. От направена статистика се знае, че ако разликата на който и да било параметър на кредита надвишава 5%, то клиентът е малко вероятно да се съгласи на новопредложеният договор.

Ако средният лихвен процент е 5% и всички клиенти, при които има голям шанс да се съгласят на новите условия, си плащат редовно вноските, то тогава само от лихвите се начисляват 939030 лева.

В крайна сметка чрез тази програма се спечелват клиенти, които иначе биха били отхвърлени. И макар и някои сметки да са по-груби, не може да се отрече, че банката ще спечели.