

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

**Кваліфікаційна робота
на здобуття ступеня магістра**

за спеціальністю 122 Комп'ютерні науки
на тему:

РОЗРОБКА СИСТЕМИ КРЕДИТНОГО СКОРИНГУ

Виконав студент 2-го курсу магістратури
Величко Дмитро Олексійович

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Кузенко Володимир Федорович



(підпис)

Засвідчую, що у цій роботі немає
запозичень із праць інших авторів
без відповідних посилань.

Студент

(підпис)

Роботу розглянуто й допущено до
захисту на засіданні кафедри теорії
та технологій програмування
« ____ » _____ 201_ р.,
протокол № ____
Завідувач кафедри
М.С. Нікітченко

(підпис)

Київ – 2021

РЕФЕРАТ

Обсяг роботи 41 сторінка, 29 ілюстрацій, 17 джерел посилань
КРЕДИТНИЙ СКОРИНГ, КРЕДИТ-СКОРИНГОВА МОДЕЛЬ,
ПРОГНОЗУВАННЯ КРЕДИТНОЇ ПЛАТОСПРОМОЖНОСТІ,
ЙМОВІРНОСТІ ЗАТРУДНЕНЬ, КОЕФІЦІЄНТИ ВАЖЛИВОСТІ, КРЕДИТНА
ЛІНІЯ, СИСТЕМА ПЕРЕВІРКИ КРЕДИТНОЇ ПЛАТОСПРОМОЖНОСТІ.

Об'єктом дослідження є процес обчислення, аналізу, прогнозування та побудови моделі кредитного скорингу, що здійснюється за вказаними параметрами клієнта, який бажає взяти кредит, за допомогою розробленої системи кредитного скорингу.

Предметом дослідження є знаходження найбільш точнішого прогнозу кредитної платоспроможності клієнтів банків

Мета роботи є розробка системи кредитного скорингу для обчислення, аналізу, прогнозу та виявлення фінансових затруднень позичальника

Методи розроблення та інструменти дослідження (розроблення): проектування системи відбувалося за допомогою UML діаграм. Система базується на багаторівневій мікросервісній архітектурі, серверна частина з бізнес логікою у свою чергу має монолітну архітектуру (n-layer architecture). Використані методи машинного навчання для побудови кредит-скорингової моделі. Інструменти розроблення: Jupyter Notebook, Python 3.7.10, Scikit-learn, seaborn, Visual Studio Code 1.45, .NET Core 3.1, MongoDB Atlas, React, Bootstrap 4, Javascript, HTML5/CSS3.

Результат роботи: виконано дослідження теоретичних аспектів кредитного скорингу, проаналізовано та досліджено кожен з параметрів позичальника та їх вплив на формування фінальної ймовірності фінансових затруднень, побудовано предиктивну кредит-скорингову модель, на базі якої встановлюються коефіцієнти важливості, розроблено програмну систему для обчислення та прогнозування кредитної платоспроможності серед позичальників.

Сфери застосування: система аналізу та прогнозування кредитного скорингу може застосовуватися у банківській сфері, а саме банками, яким потрібна класифікація на окремі групи позичальників для подальшої оцінки їх рівня кредитного ризику та кредитоспроможності.

ЗМІСТ

ВСТУП	5
1 КРЕДИТНИЙ СКОРИНГ	7
2 МОДЕЛІ КРЕДИТНОГО СКОРИНГУ	10
2.1 Дискримінантний аналіз	13
2.2 Лінійна регресія	15
2.3 Логістична регресія	15
2.4 Лінійне програмування	16
2.5 Нейронні мережі	17
2.6 Метод найближчих сусідів	20
2.7 Дерева класифікацій	21
3 СТВОРЕННЯ КРЕДИТ-СКОРИНГОВОЇ МОДЕЛІ	23
3.1 Аналіз та підготовка	23
3.2 Побудова кредит-скорингової моделі	29
4 АРХІТЕКТУРА ДОДАТКУ	31
4.1 Архітектура системи	31
4.2 Архітектура головного сервера	33
5 РОЗРОБКА СИСТЕМИ	33
ВИСНОВКИ	39
СПИСОК ЛІТЕРАТУРИ	40

ВСТУП

Оцінка сучасного стану об'єкта розробки. Зараз системи кредитного скорингу займають значне значення у житті майже кожної людини. Тому що багато людей користуються можливостями, що надають банки. З кожним роком люди частіше беруть іпотеки, позики, кредити як на щось незначне так і на більш вагомій речі. В усіх випадках, з цих операцій, банк зобов'язаний отримувати прибуток. Ефективна робота банку, в нашому випадку — видача кредитів, у значній мірі залежить у своєчасному і правильному оцінюванні позичальника, його ресурсів, його кредитоспроможності тощо. У випадку некоректного оцінювання позичальника банком, банк лише має можливість повернути кошти, у ще гірших випадках — банк отримує збитки. Саме тому кредитний скоринг використовується банками для виявлення оцінки клієнту та оцінки видачі кредиту. За допомогою кредитного скорингу банк відразу зможе встановити показник довіри до клієнта та його кредитоспроможність. Завдяки кредитному скорингу банк може уникнути незадовільних результатів під час надання кредитів. Отже, реалізація системи кредитного скорингу є актуальною проблемою на даний момент. Серед провідних альтернативних рішень в Україні можна виділити предиктивну кредит скорингову модель від Kyivstar та сервіс експрес-кредитування Loanu.

Актуальність роботи та підстави для її виконання. Головними недоліками систем кредитного скорингу є їх якість скорингу. Наприклад точність, яку показує предиктивна кредит-скорингова модель Monobank, розроблена Kyivstar, складає 48,92%, при тому що норма — 40%. Саме тому виникає необхідність в побудові та інтеграції кредит-скорингової моделі, що забезпечить більш якісні результати скорингу.

Мета й завдання кваліфікаційної роботи являє собою побудову предиктивної кредит-скорингової моделі та розроблення програмної системи розрахунку кредитного скорингу. Для досягнення цієї мети поставлено такі завдання:

- Дослідження теоретичних аспектів кредитного скорингу
- Аналіз існуючих альтернатив та програмних систем кредитного скорингу загалом
- Проаналізувати та дослідити параметри позичальника та їх вплив на формування фінальної ймовірності фінансових затруднень
- Побудувати предиктивну кредит-скорингову модель, на базі якої встановлюються коефіцієнти важливості
- Розробити програмну систему для обчислення та прогнозування кредитної платоспроможності серед позичальників

Об'єкт, методи й засоби розроблення. Об'єктом розроблення є процес обчислення, аналізу, прогнозування та побудови моделі кредитного скорингу, що здійснюється за вказаними параметрами клієнта, який бажає взяти кредит, за допомогою розробленої системи кредитного скорингу.

Були використані такі методи розроблення, як методи машинного навчання для побудови предиктивної кредит-скорингової моделі.

Для розробки кредит-скорингової системи був обраний такий інструментарій як: scikit-learn – бібліотека машинного навчання, що дає змогу створювати та тренувати різноманітні алгоритми кластеризації, класифікації та регресії, наприклад такі як random forest, лінійна регресія та градієнтний бустинг.

Можливі сфери застосування. Розроблена система аналізу та прогнозування кредитного скорингу може застосовуватися у банківській сфері, а саме банками, яким потрібна класифікація на окремі групи позичальників для подальшої оцінки їх рівня кредитного ризику та кредитоспроможності.

ОСНОВНА ЧАСТИНА

1 КРЕДИТНИЙ СКОРИНГ

Кредитний скоринг являє собою універсальний інструмент, для використання його для комплексного оцінювання різних фінансових ризиків, що визначають здатність та повну готовність позичальника сплатити кредит. Завдяки величезній кількості заяв, що надходять кожного дня та підвищеним нормативним вимогам до банків — кредит-скорингові системи стали звичайною практикою в сучасній банківській справі.

Головна суть кредитного скорингу полягає у встановленні балів певним характеристикам (параметрам) боргу та позичальникам, історичному дефолту та іншим втратам, як показник рівня ризику позичальника. Суть кредит-скорингової моделі є побудова єдиного загального показника ризику для множини факторів ризику.

Спосіб оцінки здатності виплати кредит такий як кредитний скоринг застосовується уже більше 50 років. Одним із перших успішних застосувань кредитного скорингу було використання банківських карток. Перша модель для кредитних карток роздрібного кредитного скорингу базувалася на таких параметрах:

- посада позичальника;
- роки, проведені на поточному місці роботи;
- сума щомісячного внеску.
- роки, прожиті за поточним місцем проживання;
- дані про поліси страхування життя та банківські рахунки ;
- стать людини;

Підвищення кількості банківських карток вимагало прийняття миттєвого рішення у скороченні часу. Саме тому з'явилася необхідність у розвитку та зростанні автоматизації та оптимальних методів прийняття таких рішень.

На 5 етапів можна поділити управління майбутніми кредитними

затрудненнями: управління рахунками, маркетинг, інкасація, повернення коштів та обробка заявок.

Маркетинговий етап системи показників можна використовувати декількома варіантами. Моделі реагування часто використовуються маркетинговими кампаніями, для прогнозування. Тому моделі реагування можна використовувати разом із типом моделі кредитного ризику на маркетинговому етапі, для встановлення орієнтиру лише на тих потенційних клієнтів, які, ймовірно, будуть найбільш прийнятними для маркетингового продукту [1].

За визначення кількості грошей, що будуть віддаватися позичальнику, за прийняття рішення про схвалення або відмову видачі, за умови погашення позики та за процентну ставку, яку варто нарахувати, відповідає обробка заяв. Оскільки цей процес тепер автоматизований, рішення можна приймати майже миттєво [1].

В циклі управління рахунками також використовується оцінка поведінки. Вона відрізняється від оцінок заявок тим, що вона включає характеристики, що відображають власний спосіб оплати позичальника за позику. Загальна кількість випадків, коли позичальник порушував зазначені правила, серйозність правопорушень і навіть точна дата й час коли завжди приходить оплата. Усі ці параметри прогнозують майбутню ситуацію та поведінку. Вони являють собою параметри, пов'язані з готовністю та можливістю клієнта виплатити за позику та є більш точнішими в прогнозуванні, а ніж оцінки заявок, що базовані лише на даних, що наявні на момент видачі позики.

Слід зазначити, що оцінки поведінки досить часто використовуються при змінюванні лімітів або кредитних ліній. Якщо у позичальника — низька оцінка поведінки на банківській картці, то кредитний ліміт може бути знижений банком, а також таким чином зменшиться вплив цим банком на клієнта.

Для перехресного продажу продукції використовують оцінки поведінки. Припустимо, що позикодавець має можливість використати оцінку поведінки, яка була створена для іпотечного портфеля позикодавця, для обрання клієнтів для вигідної пропозиції по банківській картці. Низька оцінка поведінки на певному продукті може завадити банку надати кредит позичальнику на інший продукт [1].

На етапах управління рахунками показники також використовуються, для допомоги утримання клієнтів. Утримування клієнтів не є таким складним процесом, як пошук нових клієнтів. Створені моделі прогнозують показник можливості виснаження для клієнтів і дозволяють розробляти нові проактивні стратегії утримання.

Важливим застосуванням систем показників є етап збору. Він використовується для того, щоб допомогти зі збором прострочених рахунків (набори на ранніх стадіях). На початку місяця зв'язуються з клієнтами з низьким рівнем збору грошових коштів, та спосіб контактування може в будь-який момент змінитися в залежності від оцінки. Зазвичай стратегії інкасації застосовують скори інкасації в комбінації з непогашеним залишком (чи сума позик за мінусом очікуваних значень застави, якщо позики забезпечені), для того щоб визначити, до кого слід звертатися і як часто [1].

Кредитні бали застосовуються для визначення плану роботи з серйозними та неправомірними рахунками (набори на пізніх стадіях). Після виявлення оцінок, що прогнозують ймовірності того, що позичальники зможуть виправитися від правопорушень, вони можуть бути застосовані для вибору позик для програм модифікації або іншого спеціального режиму.

Виділити можна ще одне застосування системи показників, а саме оцінки відновлення. Оцінки відновлення застосовуються для оцінювання ймовірності повернення абсолютно всієї або лише частини заборгованості. Вже коли позичальник порушив правила договору або навіть зняв з рахунку кошти, для ранжирування рахунків за ймовірністю, що одна з частин боргу

буде стягнута — можуть бути сформовані оцінки відновлення. Позикодавцям допомагають показники відновлення, а саме для приймання обґрунтованих рішень щодо того, які рахунки слід зберегти і які слід спробувати відновити, а які взагалі слід продати третім особам або агентствам, що займаються стягуванням боргів.

Можна підбити підсумок того, що кредитний скоринг є встановлення деякої числової оцінки потенційного клієнта, що бажає взяти кредит, спираючись на деякі його параметри, перш за все, які присутні в анкеті позичальника. Кредитний скоринг надає змогу банку:

- отримати більш точнішу оцінку позичальника;
- швидка процедура оцінки;
- мінімізація впливу людини під час аналізу;
- створення бази даних позичальників;
- понизити встановлені резерви на потенційні втрати за кредитними зобов'язаннями.

2 МОДЕЛІ КРЕДИТНОГО СКОРИНГУ

Методи штучного інтелекту (баєсові мережі, логіко-ймовірнісні методи, експертні системи, нейронні мережі, методи найближчих сусідів, генетичні алгоритми), методи статистики (лінійна регресія, логістична регресія, дискримінантний аналіз, дерева класифікацій), дослідження операцій (нелінійна оптимізація, лінійне програмування) використовують для кредитного скорингу. Ці методи зазвичай поєднуються у один єдиний метод, створюючи комбінацію моделей.

Модель будується наступним чином: беремо вибірку даних по вказаним позичальникам, як правило, їх кількість повинна бути не менше ніж кілька тисяч записів. У кожного позичальника повинна бути кредитна історія за вказаний період часу, та параметри для оцінки.

Кожна кредитна історія вирішує, чи є вона "поганою" чи "хорошою"; як правило, "поганою" кредитною історією називають ознаку того, що присутній

факт затримки платежів на чотири і більше місяців. Прийняття рішень завжди важко, щодо позичальників, які невчасно оформили платіж за два місяці. Таких клієнтів можна віднести до середньої групи. Дивлячись на те, який метод кредитного скорингу ми використали, при побудові моделі можуть бути ситуація, де всі три зазначені групи, або клієнти середньої групи — видаляються з вибірки.

Фінальна точність моделі сильно залежить від вибору періоду, за який розглядається кредитна історія (проміжок часу між отриманням характеристик позичальника і класифікацією клієнтів на "поганих" і "хороших"). Під ймовірністю дефолту будемо вважати функцію часу, яка на початку є нестабільною, і лише після одного року вона починає стабілізуватися, з цього випливає, що обрання меншого періоду може призвести до недооцінки ймовірності дефолту. Слід зазначити, що обрання проміжку, який більше двох років є не зовсім правильним рішенням, тому що можуть статися деякі зміни в економіці чи суспільстві, таким чином позичальники, які не вперше звертаються до кредитної організації, матимуть відчутну характеристичну різницю на відміну від тих, у яких вже побудована модель [2].

Співвідношення числа поганих і хороших позичальників у вхідному датасеті також здійснює вплив на якість системи кредитного скорингу. Одні методи застосовують однакові частини поганих і хороших позичальників, тоді як інші методи вимагають, щоб датасет відображав реальне співвідношення хороших і поганих клієнтів.

Стопроцентна якість класифікації — неможлива, тому що одні і ті ж набори відповідей та характеристик можуть бути представлені як поганим, так і хорошим позичальником.

Лінійна регресія та дискримінантний аналіз були найбільш використовуваними засобами побудування показникової системи. Головні переваги в тому, що ці методи є широко доступними в статистичних програмних пакетах та прості. Майже завжди числові оцінки атрибутів і

коефіцієнти поєднуються для того, щоб отримати єдині внески, які б додавалися для формування фінальної оцінки. На сьогодні частіше всього використовується логістична регресія.

Моделі штучного інтелекту та статистичні моделі є двома основними методами для побудови кредит-скорингової моделі.

Припустимо набір позик можна поділити на дві групи або класи G та B , що в нашому випадку будуть позначати позики, що після видачі виявляються поганими або хорошими. Розміри класів або груп дуже різні, так що для ймовірності того, що довільно обраний клієнт буде належати до групи G , що позначається як p_G , є $p_G > p_B$. Припустимо x буде вектором незалежних змінних (або вектором вимірювання). Цей вектор потрібен для того, щоб вирішити деякі питання про віднесення клієнта до групи B або G . Припускаємо $p(G|x)$ — ймовірність того, що позичальник із вектором вимірювання x відноситься до групи G , а для відношення до групи B ймовірність — $p(B|x)$. Скажемо, що ймовірність $p(x|G)$ показує на те, що хороший позичальник містить вектор незалежних змінних x . Отже ймовірність поганих позичальників дорівнює $p(x|B)$. Основне завдання полягає в тому, що для оцінки ймовірності $p(\cdot|x)$ з набору даних про претендентів, які виявилися хорошими чи поганими, потрібне правило про розділення простору всіх векторів незалежних змінних на групи A_B та A_G базуючись на встановлених ймовірностях, так що в A_G містяться вектори вимірювання заявників, які виявилися хорошими, або поганими. Але неможливо ідеально класифікувати, тому ще може бути ситуація, коли один і той же вектор задається двома позичальниками, де один поганий, а інший хороший. Наша задача — пошук правила, яке б мінімізувало витрати банку, а саме неправильної класифікації позичальників. Нехай c_G — витрати, які вказують на неправильну класифікацію хорошого позичальника як поганого, а c_B — витрати, що вказують на класифікацію поганого позичальника як хорошого. В більшості випадків $c_B > c_G$, адже витрати, що виникають через неправильну класифікацію

поганого клієнта, спричиняють більше фінансових проблем, а ніж витрати, що виникають через інші типи помилок. Якщо позичальників з x віднести до класу G , то прогнозовані витрати дорівнюватимуть $c_B p(B|x)$, а прогнозована втрата для всього набору даних

$$c_B \sum_{x \in A_G} p(B|x)p(x) + c_G \sum_{x \in A_B} p(G|x), \quad (2.1)$$

де $p(x)$ – ймовірність, де вектор незалежних змінних рівний x .

Мінімізується це таким чином, що до групи G призначаються такі позичальники, які мають свою групу векторів незалежних змінних

$$A_G = \{x | c_B p(B|x) \leq c_G p(G|x)\}, \quad (2.2)$$

що буде еквівалентним до

$$A_G = \{x | p(G|x) \geq (c_B / (c_G + c_B))\}. \quad (2.3)$$

Маємо змогу нормалізувати витрати, що вказують на неправильну класифікацію до $c_G + c_B = 1$ (без втрати загальності). Отже, в даному випадку, правило класифікації — віднесення позичальника з x до класу G , де $p(G|x) > c_B$, а в іншому випадку до класу B [1].

Одним з головних завдань є визначення важливості та вартості помилок у кредитуванні та якомога якісніше встановлення оптимального лімітного балу для кредитного скорингу, тому що банки зобов'язані обрати оптимальний компроміс між прибутковістю та ризиком. Політика кредитування, яка є занадто обмеженою, забезпечує мінімізацію витрат, а саме дефолтних позик, хоча схожі витрати на невидані позики можуть бути значно більшими а ніж потенційні витрати на заборгованість, з цього випливає, що прибуток не максимізується. Та навпаки, обрана політика, що буде відрізнятися, може призвести до великих збитків від заборгованості.

2.1 Дискримінантний аналіз

Основна задача лінійного дискримінантного аналізу являє собою класифікацію сукупності різних елементів на однорідні підмножини, а також подальші дії щодо цих предметів. Використання лінійного дискримінантного

аналізу базується на лінійній дискримінаційній функції, що перетинає центроїди обох класів для класифікації позичальників. Лінійна дискримінаційна функція має наступний вигляд:

$$a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n, \quad (2.4)$$

де $x_1 \dots x_n$ — змінні позичальників;

$a_1 \dots a_n$ — коефіцієнти дискримінації при n змінних позичальників.

Припустимо, що для кожного позичальника є деяка кількість пояснювальних змінних. Ціль методу — пошук лінійної комбінації пояснювальних змінних, яка б розділяла підмножини. Наприклад, візьмемо простий та зрозумілий випадок з двома підмножинами, отже мета полягає у знаходженні такої лінійної комбінації пояснювальних змінних, яка залишала б максимальну відстань між двома підмножинами.

Нехай подані багатовимірні нормальні розподіли із загальною дисперсією $p(x|B)$ і $p(x|G)$. Таким чином вищезгадане рівняння зводиться до

$$A_G = x|\sum w_i x_i > c, \quad (2.5)$$

де x_i — представляє пояснювальні змінні,

w_i — асоційовані коефіцієнти

Візьмемо $s(x) = \sum w_i x_i$ це означає, що ми можемо провести дискримінацію до цієї оцінки, а отже, таким чином, наша задача зводиться лише до одного виміру [1].

Потрібність багатовимірної нормальності — поширена помилка. Оптимальне лінійно дискримінантне правило виконується, коли змінні належать багатовимірному еліпсоїдальному розподілу.

Для побудови кредит-скорингової моделі найбільш використовуваним статистичним методом є лінійний дискримінантний аналіз.

Переваги методу:

- простий, легкий для оцінювання
- дає якісні результати, порівняно з іншими методами.

Недоліки методу:

- потреба у нормальному розподілі даних

2.2 Лінійна регресія

В кредитному скорингу лінійна регресія застосовувалась для двокласної задачі. Тому що регресія з використанням неіснуючих змінних для міток класів дає лінійну комбінацію параметрів прогнозування, яка паралельна дискримінантній функції.

Статистичний метод, який дає змогу бути використаним для аналізу взаємного зв'язку однієї або кількох незалежних змінних з однією залежною змінною — лінійна регресія. Мета лінійної регресії, за допомогою однієї незалежної змінної здійснити прогноз залежної змінної.

Ця модель застосовується для пошуку лінійної функції залежності, яка якомога якісніше передбачає значення залежної змінної як функції незалежної змінної.

2.3 Логістична регресія

Деформацією лінійної регресії будемо називати логістичну регресію. Різниця логістичної від лінійної полягає в тому, що у логістичній обмежень на гіпотези щодо даних — менше, а також логістична використовує якісні показники. Логістична регресія передбачає ймовірність дефолту позичальника і визначає змінні, що впливають на його поведінку, на відміну від лінійного дискримінантного аналізу, що аналізує кореляційність характерних змінних користувача.

Застосування логістичної регресії використовується для прогнозування ймовірності виникнення деяких ситуацій за значенням декількох ознак. Тому вводиться залежна змінна, що буде приймати лише одне з двох значень – в

кредит-скоринговій моделі, значення 1 якщо хороший позичальник та 0 якщо поганий позичальник, та множина незалежних змінних, потім за допомогою значень яких обчислюється ймовірність становлення значення залежної змінної.

Не допускаються у логістичній регресійній моделі значення залежних змінних, які менше/рівні 0, більше/рівні 1, не дивлячись навіть на значення незалежних змінних. З цього випливає, що дана модель часто застосовується для аналізування бінарних залежних змінних. Використаємо вказане нижче рівняння регресії:

$$y = e^{(b_0 + b_1x_1 + \dots + b_nx_n)} / [1 + e^{(b_0 + b_1x_1 + \dots + b_nx_n)}]. \quad (2.6)$$

В цій моделі прогнозовані значення y в усіх випадках будуть знаходитися в діапазоні від 0 до 1, незалежно від величин x чи регресійних коефіцієнтів чи величин x [6].

Задачі логістичної регресії вирішує тільки метод максимальної правдоподібності. Ідея методу — процес оцінювання регресійних коефіцієнтів, де потім потрібно, при заданих значеннях, звести все до максимізації ймовірності появи конкретної вибірки [6].

2.4 Лінійне програмування

Головна задача лінійного програмування — пошук найменшого (чи найбільшого) значення цільової лінійної функції

$$F(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (2.7)$$

при умові, що систему лінійних обмежень задовольняють x_1, x_2, \dots, x_n так, що обмеження системи можуть бути як нерівності, так і рівності.

Лінійне програмування в кредитному скорингу представлене так, припустимо маємо деякий набір даних, ці дані містять в собі відповіді на питання x_{i1}, \dots, x_{in} для кожного з N позичальників, а також індикатори, що відповідають за наявність дефолту Y_i . Спочатку виконуємо пошук функції $s(x)$,

що буде поділяти на хороших та поганих позичальників, в кожному конкретному випадку мінімізуємо помилки a_i :

$$\begin{cases} a_1 + \dots + a_N \rightarrow \min, \\ w_i x_{i1} + \dots + w_m x_{i3} \geq c - a_i, Y_i = 0, \\ w_i x_{i1} + \dots + w_m x_{i3} \leq c + a_i, Y_i = 1, \\ a_i \geq 0. \end{cases} \quad (2.8)$$

Бачимо, що по змінним $(w_1, \dots, w_n, c, a_1, \dots, a_N)$ проводиться мінімізація. Максимальну можливу помилку також можна мінімізувати, для цього потрібно замінити всі a_i на одну змінну a [6].

Переваги методу:

- можна накласти додаткові обмеження. Розглянемо приклад: кредитний продукт направлений на молодих людей, отже є можливість накласти додаткові обмеження $w_i \geq w_i^0$, як результат — скорингова функція віддаватиме перевагу людям, що відносяться до цієї вікової групи.

2.5 Нейронні мережі

Апаратні чи програмні реалізації математичних моделей, що побудовані за принципом функціонування та організації біологічних нейронних мереж називаються нейронними мережами. Розглядатися нейронні мережі можуть як метод нелінійної регресії. Але найчастіше вони використовуються для скорингу юридичних осіб, а ніж для скорингу приватних осіб. Мережа надає змогу виконувати обробку прецедентної навчальної вибірки з набагато складнішим видом сегментів.

Структура мережи включає в себе один вхідний шар, один або декілька прихованих шарів та один вихідний шар, кожен з цих шарів складається з декількох нейронів. Задача кожного нейрону — обробити свої вхідні дані та згенерувати одне вихідне значення, що буде передаватися нейронам на наступному шарі. З вектора x значення одної характеристики буде забезпечено кожним нейроном на вхідному шарі (індексовані до $i = 1, 2, \dots, n$).

Розповсюдження сигналу на усіх шарах відбувається таким чином. Для кожного нейрона обчислюється зважена сума вхідних даних: виконується множення вихідного значення цього нейрона в мережевому шарі на відповідну вагу зв'язку з поточним нейроном.

Після цього застосовується передавальна функція $g(x)$ до отриманої зваженої суми, для того щоб визначити вихідне значення нейрона. З цього випливає, що виробляється активація u у прихованому шарі (індексовані до $j = 1, \dots, g$) на кожному нейроні

$$a_j = g\left(\sum_i w_{ij} x_i\right). \quad (2.9)$$

Нейрони прихованого шару ведуть себе так само, як і нейрони вихідного шару (індексовані до $k = 1, \dots, m$), для отримання вихідних даних мережі:

$$y_k = f$$

де w_{jk} та w_{ij} – ваги.

Логістичну функцію

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

або гіперболічну тангенсну функцію

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.11)$$

прийнято використовувати для функцій f і g , у вже згаданих мережевих вихідних даних. Слід застосовувати логістичну функцію на вихідному рівні, при існуванні проблем двійкової класифікації. Структура нейронної мережі, що має прихований шар — може апроксимувати будь-яку обмежену інтегровану неперервну функцію як завгодно точно [7].

Виділяють два етапи оптимізації нейронних мереж:

- всі ваги слід ініціалізувати
- реалізація нелінійної схеми оптимізації

Спочатку ініціалізація ваг представляється будь-яким невеликим числом. Далі, після ініціалізації, здійснюється тренуванням чи навчанням нейронних мереж. Алгоритм зворотного поширення — найчастіше використовуваний алгоритм навчання багатошарових перцептронів. Ваги кожного нейрона модифікуються так, що помилки обчислення з вихідного шару, зводяться до мінімуму.

Переваги:

- відкинуті припущення, щодо взаємозв'язка між вхідними змінними
- можливість навчати мережу

Недоліки:

- немає пояснень до мережі (неможливо провести аналіз чутливості, для виділення найбільш значимих параметрів, а також пояснити прогнозування).

2.6 Метод найближчих сусідів

Головний принцип методу найближчих сусідів являє собою те, що елемент відноситься до того класу, до якого відноситься більшість сусідів цього елемента.

В основі класифікаторів k -найближчих сусідів лежить навчання за аналогією. Навчальні зразки описуються n -мірними числовими атрибутами. Кожен зразок являє собою точку в n -мірному просторі. Якщо обраний зразок — невідомий, наш класифікатор шукає, для навчальних зразків у просторі шаблонів, найближчі зразки до невідомої вибірки. Евклідову відстань визначає відстань між двома зразками:

$$d(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.12)$$

Застосовуючи цей класифікатор, важливо вибрати використану метрику. Значення k повинно бути меншим порівняно з найменшим класом.

Переваги:

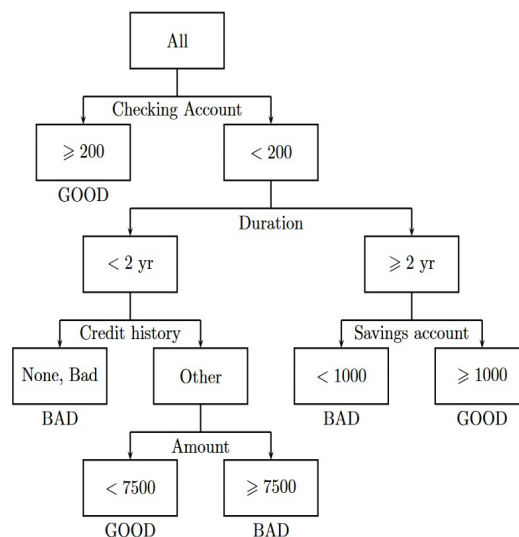
- легко навчати модель (нові дані слід просто додати до вже існуючих, можна з часом видаляти непотрібні дані, з врахуванням змін)

Недоліки:

- потреба у повному перегляді баз даних під час класифікації нового позичальника
- складнощі у виборі оптимальної метрики.

2.7 Древа класифікацій

Дані про позичальників поділимо на обмежені підмножини, для зроблення однорідності ризику дефолту вищою в підмножині, а ніж вихідні набори. Ділимо до тих пір, поки отримані підмножини не задовольняють умовам кінцевого вузла. Для побудови дерева рішень ми повинні дотримуватися трьох правил: правила, що визначають приналежність до класу кінцевого вузла, правила біфуркації та правила зупинки. Для розподілу нових підмножин використаємо правила біфуркації. Для визначення, чи є підмножина кінцевим вузлом чи ні — використаємо правила зупинки.



(Рисунок 2.7.1 – Дерево класифікації)

Дерева моделей - модифікація дерев класифікації. Ця модифікація полягає в тому, що замість невизначеної категорії — листку відповідає своя лінійна регресійна модель. За допомогою цієї моделі можна розділяти позичальників на основні групи, після цього для кожної групи використати регресійну модель.

Переваги:

- інтуїтивний,
- може мати справу з відсутніми спостереженнями
- зручна інтерпретація результатів у вигляді дерева
- можливість обробляти чисельні дані, які можуть бути мультимодальними, та категоричні предиктори з неординарною чи порядковою структурою.
- можливість легко виявляти прості взаємозв'язки між декількома змінними, які б були непоміченими іншими методами [9].

Недоліки:

- обчислювальне навантаження із-за великого набору даних
- може не мінімізувати загальні витрати на неправильну класифікацію.

Підбиваючи підсумки — не існує найкращого методу. На підбір потрібного методу впливають структури даних, деталі проблеми, цілі класифікації (зважений коефіцієнт неправильної класифікації, загальний коефіцієнт неправильної класифікації, певний показник прибутковості, низький рівень ризику серед прийнятих тощо) та використовувані характеристики. Найчастіше немає найкращого методу для різних наборів даних.

Найважливішими методами кредитного скорингу є статистичні моделі та моделі штучного інтелекту. Всі ці методи застосовуються окремо один від одного або в різних комбінаціях .

3 СТВОРЕННЯ КРЕДИТ-СКОРИНГОВОЇ МОДЕЛІ

3.1 Аналіз та підготовка

Для поліпшення оцінки кредитів на сучасному рівні, передбачивши ймовірність того, потрібно створити предиктивну кредит-скорингову модель, що визначає позичальників з фінансовими труднощами та спрогнозує їх платоспроможність на найближчі роки.

Будемо використовувати random forest класифікатор з двох причин: по-перше, тому що це дозволить нам швидко і легко змінити результат на просту двійкову проблему класифікації. По-друге, оскільки функціональність дозволяє нам вивести оцінку ймовірності, саме цю оцінку ми будемо використовувати для прогнозування ймовірності [10].

Крім того, для побудови предиктивної кредит-скорингової моделі ми переважно будемо застосовувати підхід, заснований на квантилях, щоб максимально впорядкувати процес, щоб гіпотетичні перевірені кредити могли бути повернуті якомога легше і якомога швидше [11].

Попередні етапи для побудови предиктивної кредит-скорингової моделі:

1. Пошук потрібного датасету для тренування та тесту, що містить дані про кредитну платоспроможність серед клієнтів та визначення критеріїв за якими буде здійснюватись оцінювання.

Знайдений датасет GiveMeSomeCredit має дані для тренування та тестування моделі про більш ніж 250 000 позичальників.

Розглянемо параметри, які представлені у обраному датасеті, клієнтів, що бажають взяти кредит [12]:

SeriousDlqin2yrs — особа, яка зазнала 90 днів прострочення або ще гіршого терміну

RevolvingUtilizationOfUnsecuredLines — загальний залишок за кредитними картками та особистими кредитними лініями, за винятком нерухомості та без боргів у розстрочку

age — вік позичальника у роках

NumberOfTime30-59DaysPastDueNotWorse — кількість разів, коли позичальник прострочував на 30-59 днів

DebtRatio — щомісячні виплати боргу, аліменти, витрати на життя, поділені на щомісячний валовий дохід

MonthlyIncome — місячний прибуток

NumberOfOpenCreditLinesAndLoans — кількість відкритих позик (розстрочка, як кредит на авто чи іпотеку) та кредитних ліній (наприклад, кредитні картки)

NumberOfTimes90DaysLate — кількість випадків, коли позичальник прострочував на 90 або більше днів.

NumberRealEstateLoansOrLines — кількість іпотечних кредитів та позик на нерухомість, включаючи кредитні лінії власного капіталу

NumberOfTime60-89DaysPastDueNotWorse — кількість випадків, коли позичальник прострочував на 60-89 днів

NumberOfDependents — кількість утриманців у сім'ї, крім них самих (дружина, діти тощо)

Unnamed: 0	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTime30-59DaysPastDueNotWorse	DebtRatio	MonthlyIncome
0 1	1	0.766127	45	2	0.802982	1500
1 2	0	0.957151	40	0	0.121876	2000
2 3	0	0.658180	38	1	0.085113	2500
3 4	0	0.233810	30	0	0.036050	3000
4 5	0	0.907239	49	1	0.024926	4000

(Рисунок 3.1.1 — Вигляд датасету для аналізу)

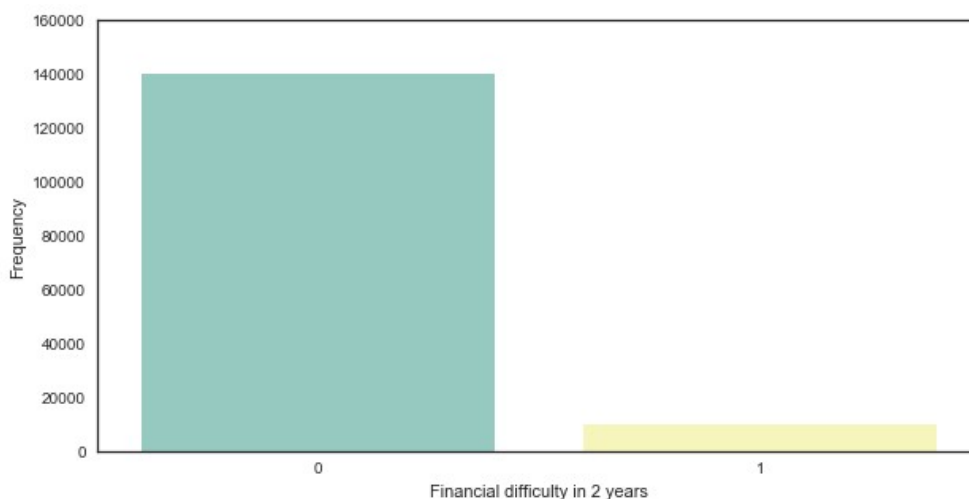
- Створення алгоритму кредитного скорингу, що буде визначати та прогнозувати ймовірності фінансових затруднень клієнта на наступні 2 роки.
- Провести аналіз частоти фінансових затруднень на наступні 2 роки та виконати цільовий розподіл (Рисунок 3.1.2) [13].

```
ax = sns.countplot(x = train.SeriousDlqin2yrs ,palette="Set3")
sns.set(font_scale=1.5)
ax.set_ylim(top = 150000)
ax.set_xlabel('Financial difficulty in 2 years')
ax.set_ylabel('Frequency')
fig = plt.gcf()
fig.set_size_inches(10,5)
ax.set_ylim(top=160000)

plt.show()
```

(Рисунок 3.1.2 — Частота фінансових затруднень на 2 роки)

Можна зробити висновок, що розподіл нашої цільової змінної є дуже відхиленням від норми (Рисунок 3.1.3).



(Рисунок 3.1.3 — Графічне зображення частоти фінансових затруднень на 2 роки)

4. Визначення результатів вимірювання, які виділяються з загальної вибірки (викиди) тренуючого набору даних (Рисунок 3.1.4)

Для визначення викидів ми проходимо по кожному параметру.

Виділяємо перший (25%) квантиль, третій (75%) квантиль та на базі цих квантилів — знаходимо міжквартильний розмах. Далі формуємо список до якого будемо додавати індекси викидів певного параметра. Вибираємо спостереження, де зустрічаються більше двох викидів.

```

def detect_outliers(df,n,features):
    outlier_indices = []

    for col in features:
        Q1 = np.percentile(df[col], 25)
        Q3 = np.percentile(df[col],75)
        IQR = Q3 - Q1

        outlier_step = 1.5 * IQR

        outlier_list_col = df[(df[col] < Q1 - outlier_step) | (df[col] > Q3 + outlier_step )].index

        outlier_indices.extend(outlier_list_col)

    outlier_indices = Counter(outlier_indices)
    multiple_outliers = list( k for k, v in outlier_indices.items() if v > n )

    return multiple_outliers

```

(Рисунок 3.1.4 — Пошук викидів)

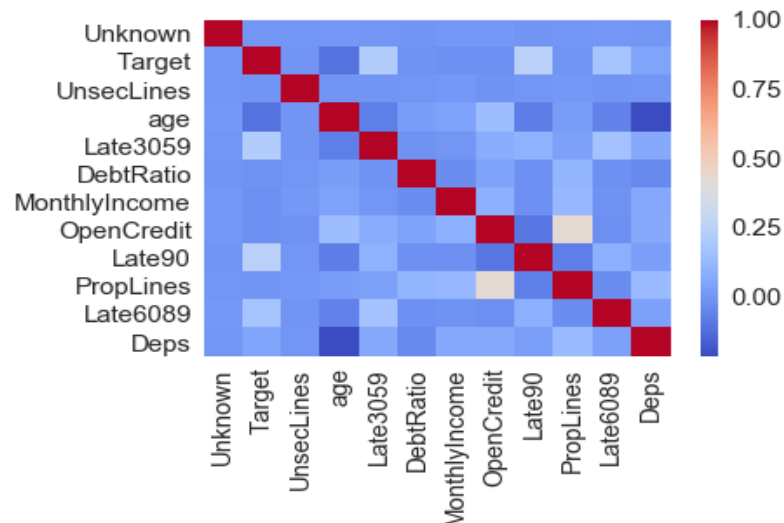
5. Розгляд параметрів з високим рівнем кореляції та побудова матриці кореляцій між оціночними параметрами (Рисунок 3.1.5)

```

: # Correlation matrix
g = sns.heatmap(train.corr(),annot=False, fmt = ".2f", cmap = "coolwarm")

```

(Рисунок 3.1.5 — Побудова матриці кореляцій)



(Рисунок 3.1.6 — Матриця кореляцій між параметрами)

Бачимо, що цільовий показник має найвищу кореляцію з віком, попередніми простроченими платежами та кількістю утриманців (Рисунок 3.1.6).

6. Дослідження та аналіз незахищених кредитних ліній та їх вплив на формування ймовірності фінансових затруднень (Рисунок 3.1.7)

```
In [55]: dataset.UnsecLines.describe()

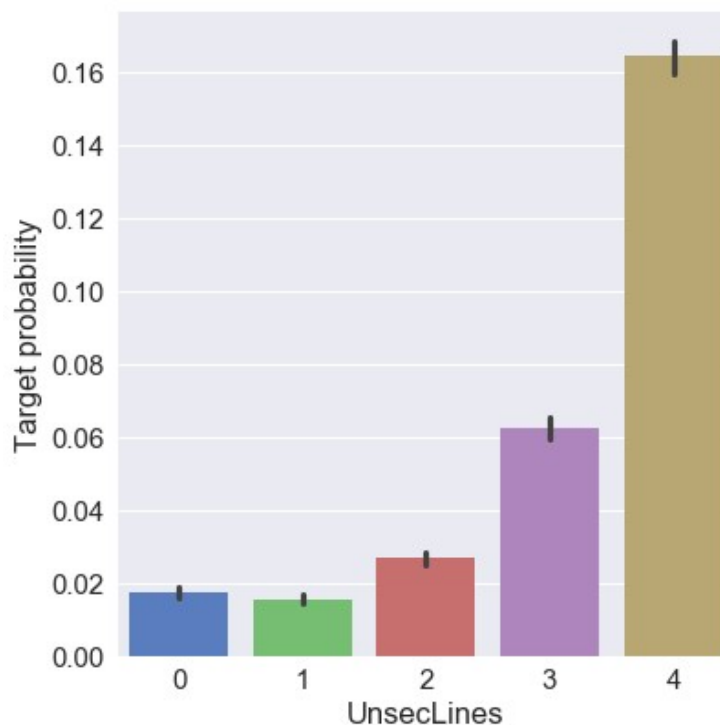
Out[55]: count    247976.000000
         mean      5.678007
         std      229.111400
         min       0.000000
         25%      0.029372
         50%      0.148802
         75%      0.544452
         max      50708.000000
         Name: UnsecLines, dtype: float64

In [56]: dataset.UnsecLines = pd.qcut(dataset.UnsecLines.values, 5).codes

In [57]: g = sns.factorplot(x="UnsecLines",y="Target",data=dataset,kind="bar", size = 6 ,
         palette = "muted")
         g.despine(left=True)
         g = g.set_ylabels("Target probability")
```

(Рисунок 3.1.7 — Дослідження незахищених кредитних ліній)

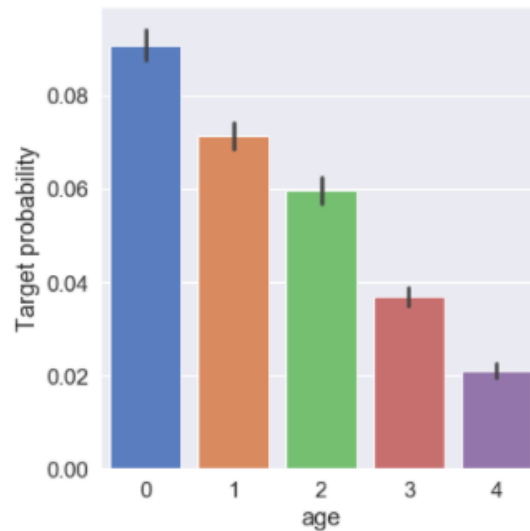
Можемо сказати, що існує майже експоненціальна залежність між цією змінною та нашою ціллю (Рисунок 3.1.8)



(Рисунок 3.1.8 — Дослідження незахищених кредитних ліній)

7. Дослідження та аналіз віку клієнтів, поділ на п'ять вікових категорій та вплив віку на формування фінальної ймовірності

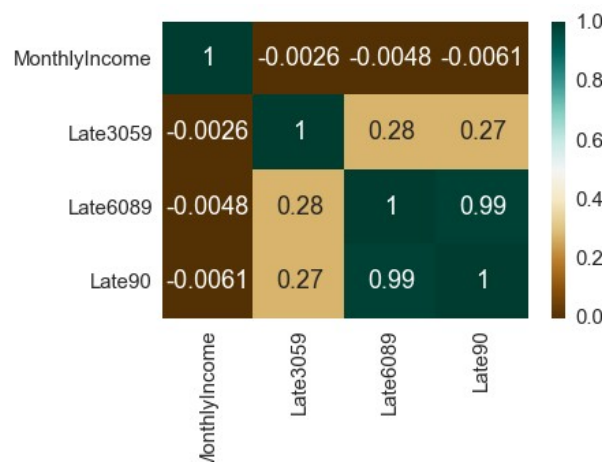
```
g = sns.factorplot(x="age",y="Target",data=dataset,kind="bar", size = 6 ,
palette = "muted")
g.despine(left=True)
g = g.set_ylabels("Target probability")
```



(Рисунок 3.1.9 — Дослідження вікових груп)

Бачимо, що вік має зворотнє відношення до ризику (Рисунок 3.1.9).

8. Дослідження та аналіз місячного прибутку клієнтів та його вплив на формування фінальної ймовірності (Рисунок 3.1.10)



(Рисунок 3.1.10— Матриця кореляцій для місячного доходу)

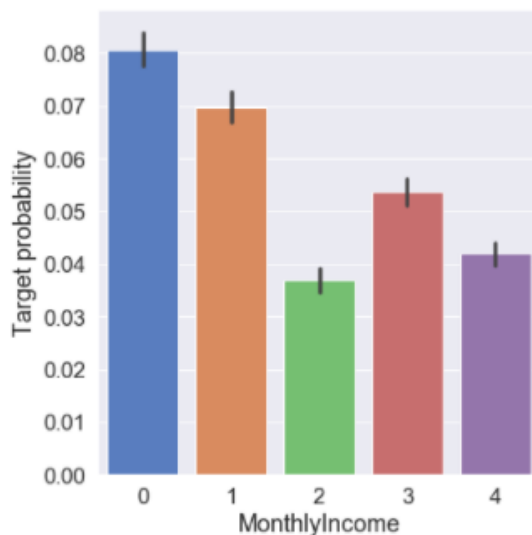
Місячний дохід не має сильної кореляції з будь-якою іншою змінною, тому ми не можемо точно оцінити значення NaN. Таким чином, ми заповнимо NaN середнім значенням та дослідимо квантилі параметру (Рисунок 3.1.11).

```
dataset.MonthlyIncome.median()
5414.0

dataset.MonthlyIncome = dataset.MonthlyIncome.fillna(dataset.MonthlyIncome.median())

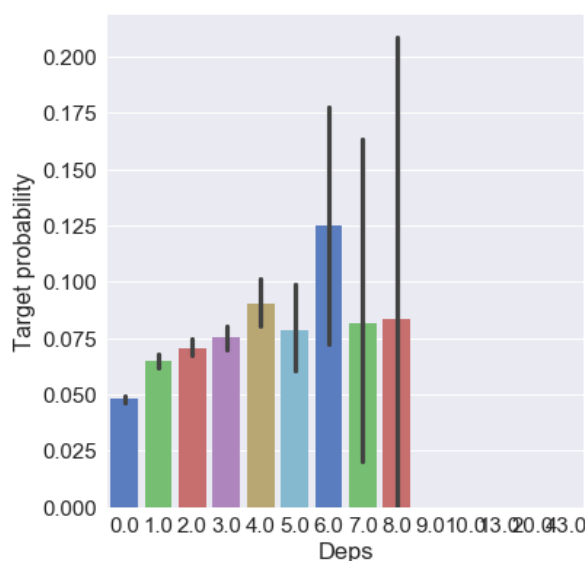
dataset.MonthlyIncome = pd.qcut(dataset.MonthlyIncome.values, 5).codes

g = sns.factorplot(x="MonthlyIncome",y="Target",data=dataset,kind="bar", size = 6 ,
palette = "muted")
g.despine(left=True)
g = g.set_ylabels("Target probability")
```



(Рисунок 3.1.11 — Аналіз місячного прибутку)

9. Дослідження та аналіз виплат місячного боргу клієнтами та його вплив на ймовірності zatrudнень (Рисунок 3.1.12).



(Рисунок 3.1.12 — Аналіз виплат місячного боргу)

3.2 Побудова кредит-скорингової моделі

Мережа базується на методі дерев рішень та random forest класифікаторі. Відбувається тренування нашої моделі та відокремлення тренуючих параметрів. Далі класифікатор random forest визначає та встановлює коефіцієнти важливості для параметрів. Результат — впорядкування за важливістю параметри (Рисунок 3.2.1).

```
In [100]: train = dataset[:train_len]
          Kaggle_test = dataset[train_len:]
          Kaggle_test.drop(labels=["Target"],axis = 1,inplace=True)

In [101]: Kaggle_test.shape
Out[101]: (101503, 55)

In [102]: train["Target"] = train["Target"].astype(int)
          Y_train = train["Target"]
          X_train = train.drop(labels = ["Target", "Unknown"],axis = 1)

In [103]: clf = RandomForestClassifier(n_estimators=50, max_features='sqrt')
          clf = clf.fit(X_train, Y_train)

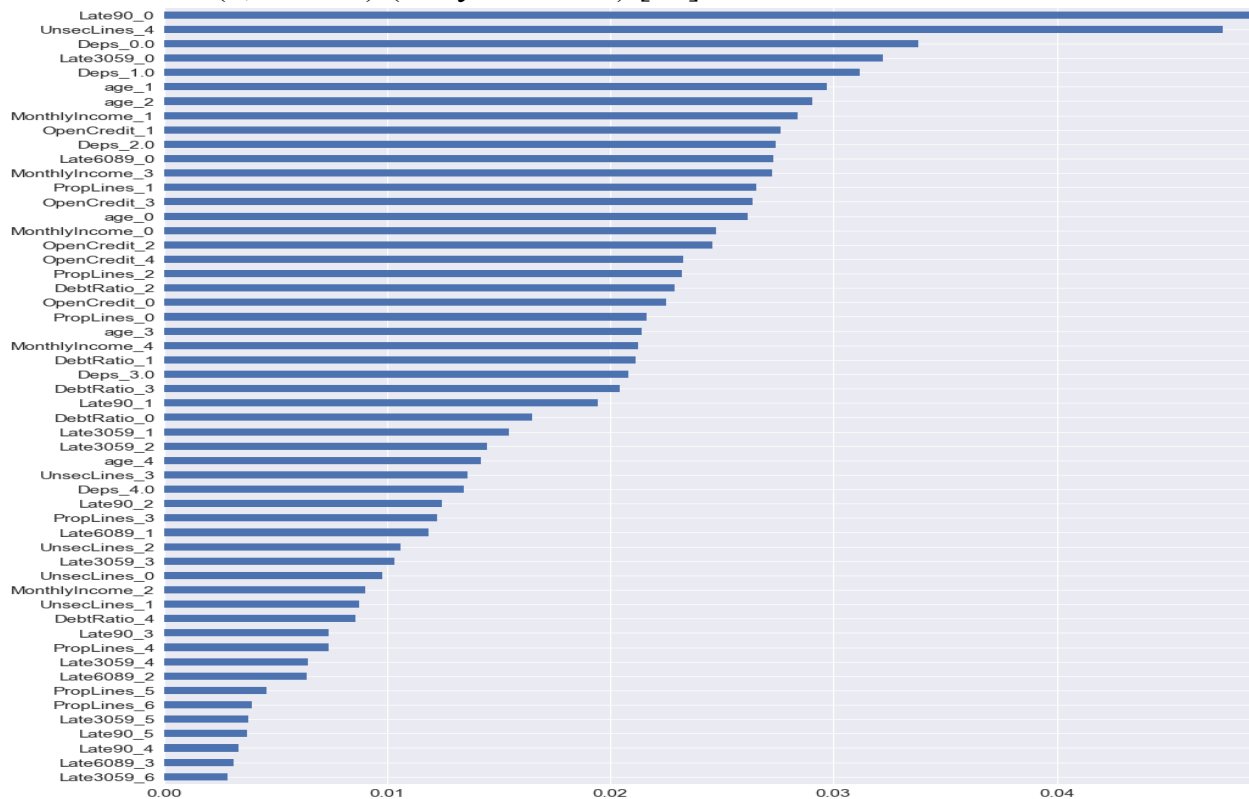
In [104]: features = pd.DataFrame()
          features['feature'] = X_train.columns
          features['importance'] = clf.feature_importances_
          features.sort_values(by=['importance'], ascending=True, inplace=True)
          features.set_index('feature', inplace=True)

In [105]: features.plot(kind='barh', figsize=(20, 20))
Out[105]: <matplotlib.axes._subplots.AxesSubplot at 0x3a5c8f1e88>
```

(Рисунок 3.2.2 — Побудова предиктивної кредит-скорингової моделі, знаходження коефіцієнтів важливості та впорядкування параметрів за важливістю)

Останнім етапом є побудова кредит скорингової моделі. Головне завдання — встановити впорядкованості оціночних параметрів за важливістю

та встановлення коефіцієнтів важливості. Точність отриманої моделі становить — (0,800498) (Рисунок 3.2.2) [14].

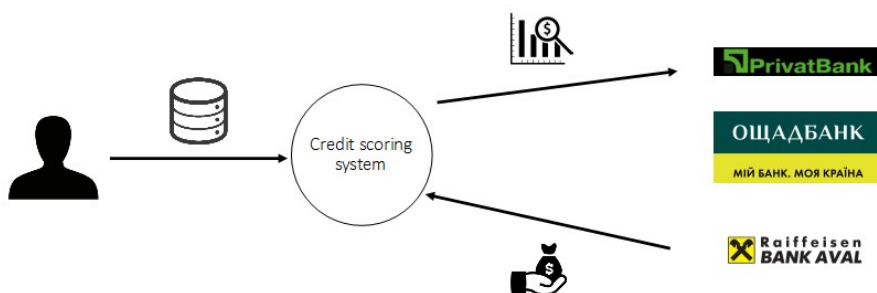


(Рисунок 3.2.2 — Впорядкування параметрів за важливістю)

4 АРХІТЕКТУРА ДОДАТКУ

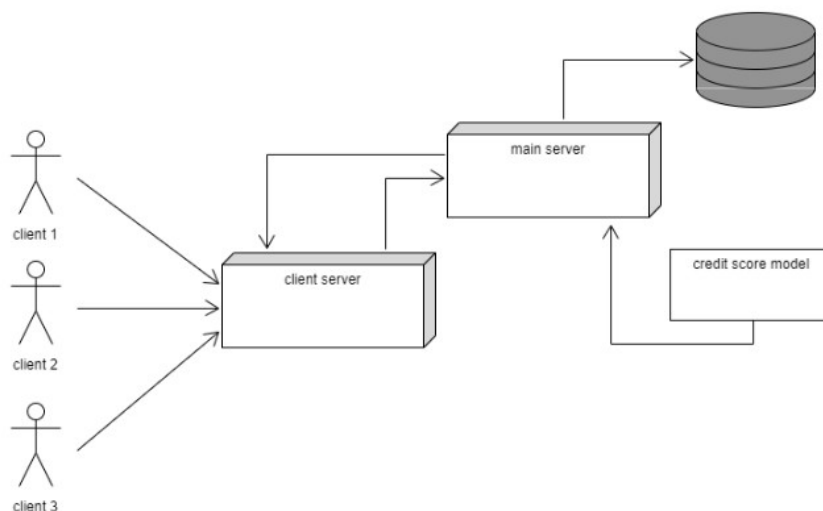
4.1 Архітектура системи

Кредит-скорингові системи зазвичай мають подібну бізнес-модель, тобто отримання даних, обробка, обчислення кредитного скору та зберігання інформації про оцінку платеспроможності позичальників банків (Рисунок 4.1.1).



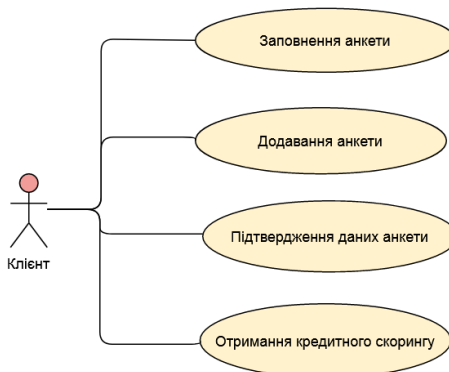
(Рисунок 4.1.1 — Бізнес-модель додатку)

Для розробки системи кредитного скорингу була обрана мікросервісна архітектура. Система складається з клієнта, основного сервера та сервера на якому будується предиктивна кредит-скорингова модель (Рисунок 4.1.2). Кожна частина додатку являє собою незалежні складові децентралізованої системи. Головний сервер базується на n-layer архітектурі [15].



(Рисунок 4.1.2 — Архітектура додатку)

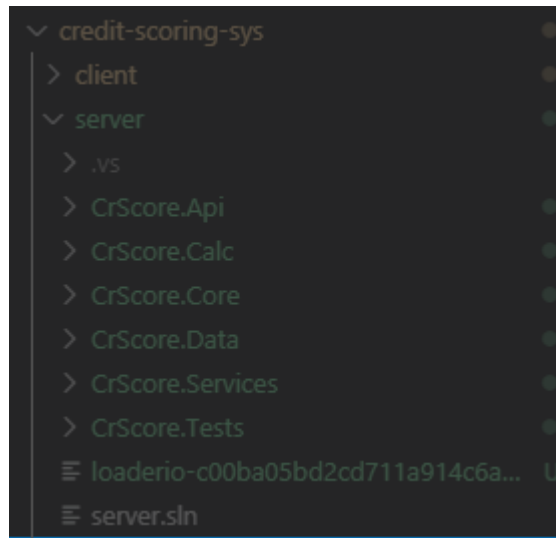
Дана діаграма показує можливі варіанти використання програми управляючим системою. Кредит-скорингова система надає змогу банку заповнювати анкету клієнта, підтверджувати ці дані та базуючись на них — отримувати кредитний бал, що показує чи позичальник задовольняє вказаним вимогам, щодо надання кредиту. Також на базі цього скору, банк приймає рішення про надання або відмову у видачі кредиту [16].



(Рисунок 4.1.3 — Діаграма прецедентів для працівника банку)

4.2 Архітектура головного сервера

Головний сервер представляє собою web арі додаток, який базується на багаторівневій архітектурі. Він поділяється на окремі проекти, які скріплені посиланнями один на одного, кожен проект відображає в собі певний рівень, що виконує тільки свої обов'язки (Рисунок 4.2.1) [17].



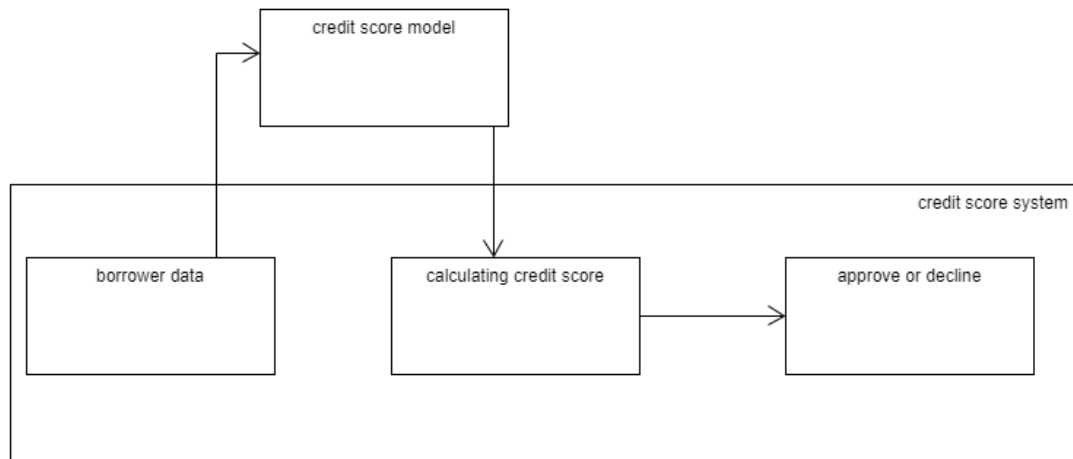
(Рисунок 4.2.1 — Структура головного сервера)

- Core – рівень в якому зберігаються відомості про моделі та контракти (інтерфеси та інше)
- Data – рівень в якому ми маємо доступ до даних та можемо виконувати різні маніпуляції з даними
- Service – рівень для реалізації бізнес логіки проекту
- Api – рівень, де виконується доступ до нашого додатка
- Calc – окремий рівень будування кредит-скорингової моделі
- Tests – окремий рівень для інтегрованого та mock тестування

5 РОЗРОБКА СИСТЕМИ

Робота системи буде виглядати наступним чином: на вхід отримуємо клієнтську інформацію з відповідними параметрами позичальника, отримані дані пропускаєм через нашу предиктивну кредит-скорингову модель, що віддає нам у відповідь ймовірності фінансових затруднень у майбутньому та коефіцієнти важливості. Кредит-скорингова система приймає коефіцієнти

важливості для того, щоб вона обчислила бали для вибраного позичальника. В кінці приймається рішення — видавати кредит чи ні (Рисунок 5.1).



(Рисунок 5.1 — Приклад процесу роботи системи)

Розробка розподіленої предиктивної системи кредитного скорингу вимагає виконання деяких передумов: постановка задачі, аналіз існуючих альтернатив, визначення особливостей кожної з них, проектування системи, вибір архітектури додатка, створення ефективного алгоритму побудови кредит-скорингової моделі для досягнення якісного кредитного скору (>40%). Після виконання цих етапів можливий перехід до розробки.

1. Побудова структури проекту (Рисунок 4.2.1)
2. Побудова предиктивної кредит-скорингової моделі (така, що здатна передбачити та спрогнозувати безпосередньо під певний сегмент).
(Рисунок 3.2.2)
3. Налаштування інтерфейсу взаємодії між програмами
4. Розробка клієнтської частини, яка буде виконувати наступний функціонал:
 - виконувати запити до серверної частини додатка (Рисунок 5.2)

```

23   public static getScores(
24     applicantInfo: IApplicantInfo
25   ): Promise<IScores[]> {
26     return axios
27       .get(`https://localhost:44347/api/${applicantInfo.score}`)
28       .then(response => response.data)
29       .then(scores => array(scoreDecoder).runWithException(scores));
30   }

```

(Рисунок 5.2 — Запит на отримання кредитних балів)

Для відправлення запиту з клієнтського сервера на основний сервер — ми використали axios, за допомогою якого можна з легкістю виконувати http запити, присутня підтримка Promise API, автоматично трансформує дані запиту та відповіді на запит у зручний JSON формат.

Метод getScores () здійснює запит до іншого сервера, який в свою чергу дасть відповідь у вигляді списку балів для певного позичальника.

оновлювати бал довіри після кожного перерахунку (Рисунок 5.3)

```

84   updateScore = () => {
85     const DEBT_FACTOR = -1
86     const DAYS_SINCE_LAST_MISSED_CUTOFF = 90
87
88     var list = this.state.list.getScores();
89
90     var completenessWeights = list.map(i => i.ci.kind == "DEBT" ? 5 : 1)
91     var activeWeights = completenessWeights.map((c, i) => list[i].active?c:0)
92     function sum(a) {
93       return a.reduce((x,y) => x + y, 0)
94     }
95     var completeness = sum(activeWeights) / sum(completenessWeights)
96     this.setState({ score, completeness })
97   }

```

(Рисунок 5.3 — Зображення оновлення кредитного скору з клієнської сторони)

Кожен раз після встановлення коефіцієнтів та обрахунку скору — здійснюється виклик методу getScores () та оновлення стану скору на сторінці.

- відобразити лістинг позичальників, який буде наочно демонструвати основні відомості клієнтів банку (Рисунок 5.4)

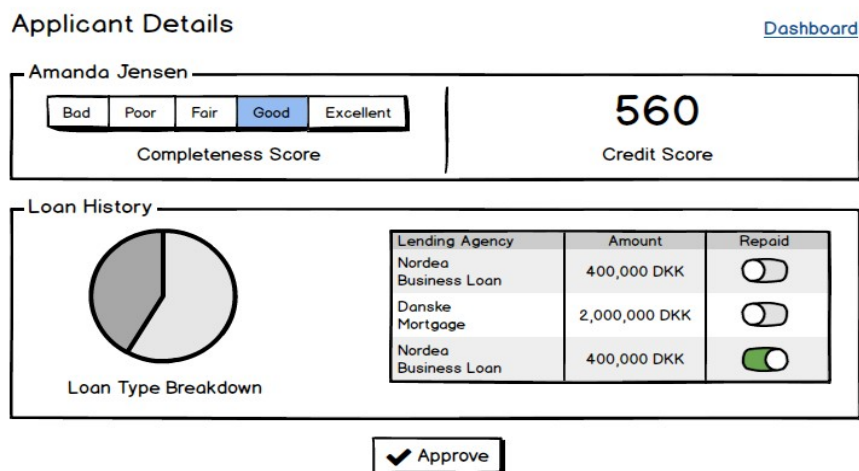
Loan Applicants [Dashboard](#)

Loan Type Q search

Loan Applicant	Loan Type	Amount	Customer	Completeness Score	Credit Score	
Amanda Jensen Copenhagen, Denmark	Mortgage	2,000,000 DKK	<input checked="" type="checkbox"/>	Good	560	
Christian Eriksen Middelfart, Denmark	Business	550,000 DKK	<input type="checkbox"/>	Poor	450	
Kyle Lafferty Belfast, UK	Mortgage	2,250,000 DKK	<input type="checkbox"/>	Excellent	750	
Nicklas Bendtner Copenhagen, Denmark	Personal	35,000 DKK	<input checked="" type="checkbox"/>	Poor	400	
Andreas Christensen Lillerød, Denmark	Student	100,000 DKK	<input checked="" type="checkbox"/>	Fair	510	

(Рисунок 5.4 — Лістинг позичальників)

- модуль, що показує детальну інформацію про позичальника. В модулі присутнє ім'я клієнта, бал довіри, історія, бал компетентності (Рисунок 5.5)



(Рисунок 5.5 — Модуль інформації про позичальника)

- додавання можливості обчислення фінальної оцінки

Після того як система отримала всі потрібні дані пройшли перевірку. Вона починає вираховувати кредитний бал клієнта враховуючи всі дані, що надає клієнт. Деякі параметри позичальника, що не були підтвердженими — не враховуються при обчисленні кредитного скорингу. Кожен не підтверджений параметр знижує бал кредитного скорингу. (Рисунок 5.6)

Credit Data	Value	Revealed
Birth year	1990	<input checked="" type="checkbox"/>
Payroll account	3,227\$	<input checked="" type="checkbox"/>
Payslips	33,523\$	<input checked="" type="checkbox"/>
Debt report	1,000,000\$	<input checked="" type="checkbox"/>
Assets	1,100,000\$	<input checked="" type="checkbox"/>
Pension report	200,000\$	<input checked="" type="checkbox"/>
Debt history	100 days since last missed payment	<input checked="" type="checkbox"/>

(Рисунок 5.6 — Приклад результату підрахунку кредитного балу)

Клієнтська сторона модуля можливості оцінки для певного клієнта, що бажає взяти кредит була реалізована за допомогою react-bootstrap, що дозволяє використовувати компоненти bootstrap але у представленні react компонента, що значно поліпшує використання bootstrap в поєднанні react та typescript.

(Рисунок 5.7)

```

120 <Navbar>
121   <Navbar.Header>
122     <Navbar.Toggle />
123   </Navbar.Header>
124   <Nav>
125     <NavItem>Add data source</NavItem>
126   </Nav>
127   <Nav pullRight>
128     <Navbar.Text className="ScoreText">Score {this.state.score}</Navbar.Text>
129     <Navbar.Text>Completeness {Math.round(this.state.completeness * 100)}%
130       <meter value={this.state.completeness} min="0" max="1">
131         {Math.round(this.state.completeness * 100)}</meter>
132     </Navbar.Text>
133     <Navbar.Text>{this.props.data.name}{this.state.age ?
134       ` (${this.state.age})` :
135       ""}
136   </Navbar.Text>
137 </Nav>
138 </Navbar>

```

(Рисунок 5.7 — Відображення скору)

Вказані параметри таблиці інформації про позичальника отримуються із підкомпонента CreditInfo bootstrap-react компонента таблиці (Рисунок 5.8).

В компоненті CreditInfo вказані платіжні відомості клієнта про його історію транзакцій (Рисунок 5.9)

```

139 <Table striped bordered condensed hover>
140 <thead><tr><th>Credit Data</th>
141 <th>Value</th>
142 <th>Revealed</th></tr></thead>
143 <tbody>
144 {
145   this.state.list.map(({ ci, active }, i) =>
146     <CreditInfo key={i}
147       creditInfo={ci}
148       handleActivityChange={(ci, active) => this.setActive(ci, i, active)}
149       active={active}
150     />)
151   }
152 </tbody>
153 </Table>

```

(Рисунок 5.8 — Перелік параметрів позичальників та їх значень)

```

54 <td className="dataColumn">
55 <Modal show={this.state.showModal} onHide={this.close}>
56 <Modal.Header closeButton>
57 <Modal.Title>Payroll Account Transactions</Modal.Title>
58 </Modal.Header>
59 <Modal.Body>
60 <Table striped bordered condensed hover>
61 <thead><tr><th>Amount</th><th>Category</th><th>Date</th></tr></thead>
62 <tbody>
63 {
64   this.state.accountData.map((transaction,i) =>
65     <tr key={i}>
66       <td className="dataColumn">{transaction.amount}</td>
67       <td>{transaction.category}</td>
68       <td>{transaction.date}</td>
69     </tr>
70   )}
71 </tbody>
72 </Table>
73 </Modal.Body>
74 <Modal.Footer>
75 <Button onClick={this.close}>Close</Button>
76 </Modal.Footer>
77 </Modal>
78 {this.props.creditInfo.data.url ? <a onClick={() => this.showUrlInModal(this.props.creditInfo.data.url)}>
79   {this.getDataString(this.props.creditInfo.data)}
80 </a>
81 : this.getDataString(this.props.creditInfo.data)}
82 </td>

```

(Рисунок 5.9 — Модуль платіжних відомостей клієнта)

5. Розробка серверної частини

Як вже було зазначено раніше — головна ідея серверної частини є отримання від предиктивної кредит скорингової моделі коефіцієнтів важливості (або скорів) та передача їх до клієнтського серверу.

Для цього було прийнято рішення розробки web арі, до якого буде звертатися клієнт.

Асинхронний метод GetAllScores() робить запит до бази даних MongoDB Atlas, для отримання колекції скорів позичальників та формує їх у нову структуру відповіді, задля коректного вигляду даних на клієнті (Рисунок 5.10).

```

35 [HttpGet("{query}")]
36 public async Task<ActionResult<IEnumerable<Score>>> GetAllScores(string query)
37 {
38     IEnumerable<Score> scores = new MongoClient(MongoConfig.connectionString).GetDatabase(MongoConfig.dbName)
39     .GetCollection<Score>(MongoConfig.collection)
40     .GetAllScoresAsync();
41
42     if (scores == null) return Content("scores not found.");
43
44     ApiResponse response = new ApiResponse()
45     {
46         totalCount = scores.Count(),
47         scores = scores.ToList()
48     };
49
50     return Ok(response);
51 }

```

(Рисунок 5.10 — Асинхронний web API метод отримання всіх скорів позичальників)

Колекція даних у MongoDB Atlas про параметри клієнта, що бажають взяти кредит виглядає наступним чином (Рисунок 5.11)

```

    _id: ObjectId("60857271df81e0cd45bd8167")
    Name: "bob"
    SeriousDlqin2yrs: 1
    RevolvingUtilizationOfUnsecuredLines: 0.766126609
    age: 45
    NumberOfTime30-59DaysPastDueNotWorse: 2
    DebtRatio: 0.802982129
    MonthlyIncome: 9120
    NumberOfOpenCreditLinesAndLoans: 13
    NumberOfTimes90DaysLate: 0
    NumberRealEstateLoansOrLines: 6
    NumberOfTime60-89DaysPastDueNotWorse: 0
    NumberOfDependents: 2

```

(Рисунок 5.11 — Параметри позичальника у MongoDB Atlas)

ВИСНОВКИ

Головними недоліками багатьох кредит-скорингових систем є їх складність, не зовсім якісні показники скорингу, а також залежність від даних на яких тренується кредит-скорингова модель. Саме тому виникає необхідність в розробці предиктивної кредит-скорингової системи, яка давала б більш точніші прогнози, спираючись на певний сегмент даних, що забезпечить ефективний аналіз, прогноз та обчислення кредитного балу.

Результатом роботи є виконання таких пунктів:

- дослідження теоретичних аспектів кредитного скорингу
- загальний огляд кредит-скорингових систем та сучасних методів для створення та обчислення кредитного скорингу
- аналіз переваг та недоліків альтернативних систем
- проаналізовано та досліджено кожен з параметрів позичальника та їх вплив на формування фінальної ймовірності фінансових затруднень
- побудовано предиктивну кредит скорингову модель, на базі якої встановлюються коефіцієнти важливості
- розроблено програмну систему для обчислення та прогнозування кредитної платоспроможності серед позичальників
- досягнуто високої точності кредит скорингової моделі

Отримані результати роботи (кредит скорингова система для аналізу, прогнозування, обчислення та побудови предиктивної кредит скорингової моделі) можуть застосовуватися у банківській сфері, а саме банками, яким потрібна класифікація на окремі групи позичальників для подальшої оцінки їх рівня кредитного ризику та кредитоспроможності.

СПИСОК ЛІТЕРАТУРИ

1. Болтон К. Логістична регресія та її застосування в кредитному скорингу – 2009 – 240 с.
2. Сяо-Лінь Лі, Ю Чжун. Огляд особистого кредитного скорингу: методи та майбутня робота – 2012 – 189 с.
3. Валютне управління Сінгапуру. Вказівки з практики управління ризиками – кредитний ризик – 2013 – 20 с.
4. Гумпарті С., Хатрі С. and Манічкавасагам В.. Дизайн і розробка моделі кредитного рейтингу для банків державного сектору в Індії – 2011 – 124 с.
5. StatSoft. Фінансові установи та статистика. Тематичне дослідження: кредитний скоринг – 12 с.
6. Дюран Д., Елементи ризику при фінансуванні споживчих платежів.
7. Томас Л. К., Огляд кредитного та поведінкового скорингу: прогнозування фінансового ризику кредитування споживачів – 2000 – 172 с.
8. Лю Ю. Оцінка класифікаційних моделей для кредитного скорингу – 2002.
9. Андерсон Р. Набір інструментів для оцінки кредитоспроможності: теорія та практика управління ризиками роздрібної торгівлі та атоматизація рішень – 2007.
10. Войтек М., Комченда Е.. Методи кредитного скорингу – 2005.
11. Онг М.К.. Кредитний рейтинг, методології, обґрунтування та ризик дефолту – 2002.
12. GiveMeSomeCredit [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com/c/GiveMeSomeCredit/data>
13. Модели кредитного и поведенческого скорингу – 27 с.

14. BaseGroup Labs. Скоринговые модели для оценки кредитоспособности заемщиков – 41 с.
15. Web App Architectures [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azurecommon-web-application-architectures>
16. Лапыгин Ю. Н. Управление проектами: от планирования до оценки эффективности. – М.: Омега-Л, 2018. – 252 с.
17. .NET Microservices Architecture [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/>