МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

НА ТЕМУ

Інтелектуальна інформаційна система визначення кредитоспроможності позичальника

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Аналітика даних»**

Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи АнД-41

Бовсуновська М.Є. _____
(прізвище та ініціали)

Керівник Мінаєва Ю. І _____
(прізвище та ініціали)

к.т.н, доц. _____
(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту рішенням кафедри *інтелектуальних технологій*
Протокол №11 від 06.06.2022 р.
зав. кафедри _____ доц. Іларіонов О.Є.

Київ – 2022

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра інтелектуальних технологій

Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.
_____
"___"_____2022 р.

ЗАВДАННЯ

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Бовсуновській Марії Євгенівні
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи)
   Інтелектуальна інформаційна система визначення кредитоспроможності позичальника
   затверджена протоколом засідання кафедри від «23» грудня 2021 р. № 4
2. Термін здачі студентом закінченого проекту (роботи) 29 травня 2022 року
3. Вихідні дані до проекту (роботи)
   Набір даних, який класифікує людей, що описуються набором атрибутів, як хороші чи погані кредитні ризики.
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
   Дослідження та аналіз предметної області, що пов'язана із системами оцінки кредитоспроможності, програмна реалізація інтелектуальної інформаційної системи визначення кредитоспроможності позичальника.
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
   Мета дослідження дипломного проекту (1 слайд), актуальність завдання (1 слайд), контекстна діаграма програмного модулю розробки (1 слайд), опис предметної області (2 слайди), постановка задачі (1 слайд), функціональний аналіз (1 слайд), опис навчального набору даних (1 слайд), алгоритм вирішення задачі (1 слайд), схема архітектури програмного модулю та опис інструментів програмної реалізації (1 слайд), оцінка результатів та приклади роботи застосунку (2 слайди), висновки (1 слайд).

6. Консультанти з випускної кваліфікаційної роботи із зазначенням розділів випускної кваліфікаційної роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 15 лютого 2022 року

Керівник _____ / Мінаєва Ю.І. /
(підпис) (ПІБ)

Завдання прийняв до виконання _____ / Бовсуновська М.Є. /
(підпис) (ПІБ)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1.	Обговорення з керівником постановки завдання, підбір літератури, огляд існуючих рішень	15.02.2022 – 01.03.2022	
2.	Аналіз постановки задачі, формалізація задачі, аналіз літературних джерел, написання розділу 1	02.03.2022 – 20.03.2022	
3.	Проектно-технологічна реалізація інтелектуальної інформаційної системи визначення кредитоспроможності позичальника.	21.03.2022 – 11.04.2022	
4.	Розробка та тестування інтелектуальної інформаційної системи визначення кредитоспроможності позичальника	12.04.2022 – 15.05.2022	
5.	Робота над оформленням пояснювальної записки та презентаційних матеріалів	16.05.2022 – 29.05.2022	

Студент-дипломник _____ / Бовсуновська М.Є. /
(підпис) (ПІБ)

Керівник випускної кваліфікаційної роботи _____ / Мінаєва Ю.І. /
(підпис) (ПІБ)

АНОТАЦІЯ

Бовсуновська Марія Євгенівна виконала випускну кваліфікаційну роботу на тему «Інтелектуальна інформаційна система визначення кредитоспроможності позичальника» за спеціальністю 122 – «Комп'ютерні науки».

У випускній кваліфікаційній роботі проведено аналіз сучасних методів оцінки кредитоспроможності позичальників, розроблено інформаційне та програмне забезпечення, що проводить процедуру оцінки кредитоспроможності позичальника.

Ключові слова: кредитування, оцінка, інтелектуальна система, нейронні мережі.

SUMMARY

The degree project: «Intelligent information system for determining the creditworthiness of the borrower» has completed by **Bovsunovska Mariia** specialty 122 – «Computer Sciences».

In this graduation thesis the modern methods of assessing the creditworthiness of borrowers, developed information and software that conducts the procedure for assessing the creditworthiness of the borrower.

Keywords: crediting, evaluation, intelligent system, neural networks.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

НБУ - Національний Банк України

NPL - non-performing loans (частка непрацюючих кредитів)

TP - true-positive (правильно розпізнано клас 1)

TN - true-negative (правильно розпізнано клас 2)

FP - false-positive (помилка першого роду)

FN - false-negative (помилка другого роду)

БД - база даних

SQL - structured query language (мова структурованих запитів)

SVM - Support-vector machine (метод опорних векторів)

IQR - interquartile range (міжквартильний розмах)

ROC - curve - receiver operating characteristic curve (крива робочої характеристики приймача)

AUC - area under curve (область під кривою)

SMOTE - Synthetic Minority Oversampling Technique (техніка над-генерації синтетичної меншості)

ВСТУП

Темою даної роботи є написання дипломної роботи за темою «Інтелектуальна інформаційна система визначення кредитоспроможності позичальника». Використання аналізу даних дозволить виявити невидимі закономірності і значно покращити процес прийняття рішення про надання кредиту. Проте варто пам’ятати, що не існує універсальних методів аналізу або алгоритмів, що придатні для обробки будь-яких типів інформації.

Метою даної роботи є створення інтелектуальної інформаційної системи, яка проводитиме скорингову оцінку платоспроможності позичальника за отриманими від нього даними та, після вирішення задачі класифікації, надаватиме рекомендації користувачу веб-застосунку як фінансово-кредитній установі про погодження чи відмову на підписання кредитного договору.

Об’єктом дослідження є надання рекомендацій для якнайменш збиткового прийняття рішення щодо видачі позики

Предметом дослідження являються методи та засоби автоматизованого визначення кредитоспроможності позичальника.

Більшість питань, що задаються людьми, пов’язані з проблемами, які вони намагаються якомога чіткіше окреслити, і, врешті-решт, керуючись питаннями, побудувати шлях до їх вирішення. В даному випадку, уявимо таку гіпотетичну ситуацію: Людина має офіційне працевлаштування, велику заробітну плату і не має жодних проблем із законом. Начебто не є проблемою отримання кредиту готівкою або підвищення ліміту на кредитній карті у знайомому банку. Але іноді, навіть у дуже надійного та чесного на перший погляд позичальника можуть виникнути труднощі з оформленням кредиту.

У першому розділі даної роботи, будуть розглянуті варіанти, за яких причин гіпотетичний клієнт міг потрапити у проблемну ситуацію відмови у наданні позики, методики та проблеми оцінки кредитоспроможності позичальника, а також, яким чином на це впливає конкуренція між фінансово-кредитними установами. Будуть описані існуючі рішення та сформовано детальну постановку задачі для даної проблеми.

У другому розділі даної роботи буде детально оглянуто структуру набору даних, сформовано короткі теоретичні відомості про методи аналізу даних, які будуть використані у роботі. Буде проведено планування та проектування архітектури системи оцінки кредитоспроможності позичальника, проведено детальний функціональний та процесний аналіз, наведено етапи розробки інтерфейсу веб-додатку.

У третьому розділі даної роботи буде описано особливості програмної реалізації мети проекту - обґрунтовано вибір інструментів реалізації та параметрів моделей використаних алгоритмів, описано структуру програмного забезпечення та базу даних користувачів. Наведено приклади роботи веб-застосунку.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ «СПОЖИВЧЕ КРЕДИТУВАННЯ» ТА ПОСТАНОВКА ЗАДАЧІ КЛАСИФІКАЦІЇ

1.1. Конкуренція підприємств надання банківських послуг

При заснуванні власного бізнесу кожен підприємець завжди має бажання отримувати великі прибутки незалежно від того, чи то приватний підприємець, чи то державна установа. Масове споживче кредитування в останні часи почало набирати популярність, як вид діяльності, що потенційно приносить досить гарні прибутки, адже громадяни України все більше звикають до покупок у розстрочку. Звиклість до комфортного проживання та відсутності заборгованостей за комунальні послуги - це те, що стимулює розвиток в країні споживчого кредитування як способу заробітку. Споживче кредитування - це видача позики певній приватній особі на витрати, що не пов’язані з комерційною діяльністю. Умовно воно ділиться на два класи, за потребою: купівля певних товарів у розстрочку та гроші з коротким терміном на задоволення власних потреб до часу, коли буде нарахована заробітна плата, іншими словами, - мікропозика (не плутати з мікрокредитуванням, яке надається для фінансування підприємницької діяльності). Розглядатимемо банки та небанківські кредитні установи, такі як наприклад кредитні спілки, ломбарди, лізингові компанії і тд. Галузь надання банківських послуг в Україні - це мінлива, нестабільна галузь, в якій конкуренція досить сильно тисне на всіх. [1] Відповідно до Закону України про банки та банківську діяльність [2], НБУ (Національний банк України) зобов’язаний тримати під наглядом та здійснювати контроль за ліцензованими небанківськими установами. На початок 2022 року в Україні зареєстровано 71 діючих банка.[3] Серед небанківських установ, станом на початок лютого 2022 року,

зареєстровано 1 довірче товариство, 275 кредитних спілок, 259 ломбардів, 949 фінансових установ, 135 лізингодавців (особи, що не являються фінансовими установами, але фактично мають право надавати послуги лізингу). [4]

Конкуренція - один основних термінів, що підтримує ринкову економіку як таку. В останні роки сильно помітне зростання рівня конкуренції в Україні. При чому, візьмемо до порівняння пострадянський простір, адже це частина нашої історії. В ньому можна побачити яскравий приклад відсутності конкуренції, через те, що комерційні установи як такі були не спроможні конкурувати з основними, державними. [5,6]

“Конкурентоспроможність банку - це здатність фінансової установи вести ефективно господарські діяльності та мати на меті досягнення практичної прибуткової реалізації послуг в умовах конкурентного ринку. Водночас створення та реалізація конкурентоспроможних послуг є узагальнюючим показником стійкості банку на фінансовому ринку у процесі ефективного використання фінансового, науково-технічного та кадрового потенціалу”. [6,7] Саме тому прийняття кращих чи інноваційних рішень в управлінні підприємством чи роботі з клієнтами, взагалі, у будь-якій області надання банківських послуг, може мати вирішальне значення - чи буде продовження цієї справи релевантним і наскільки конкурентоспроможним та стійким на ринку буде підприємство.

Прикладами реалізації унікальних ідей створення та функціонування банківської установи можуть слугувати Світовий банк та Міжнародний валютний фонд. Саме через те, що це перші установи такого роду, що були сформовані аж у 1945, їх важливість не викликає сумнівів. Їх конкурентоспроможність, як найбільших представників займаної ніші, стабільна. Основна різниця між ними у тому, що Світовий банк загалом сфокусований на фінансовій допомозі бідним країнам, покращенні стану

життя їх населення, а також поступовому сприянню зростанню стійкості глобалізації. Міжнародний валютний фонд зосереджений на відносно короткострокових кредитах, що можуть бути надані будь-яким країнам, що є членами цієї установи, і проходять через нелегкий етап економічної кризи. Але навіть подібного, міжнародного, рівня організації не мали б сенсу існування, якби не мали певної власної системи оцінки кандидата на доцільність чи пріоритетність видачі фінансової підтримки, який простіше визначити як значний за розмірами кредит.

Важливо згадати, що нечесна конкуренція, а окрім того, введення клієнта в оману, все ще має місце в Україні, хоча Закон про споживче кредитування та його правки, що набувають чинності в останні роки, намагаються все ж контролювати та попереджати подібні випадки. Постанови додають жорсткіші правила для банків, небанківських кредитних установ та осіб, що мають право надавати фінансові послуги. Правила стосуються великої кількості деталей у оформленні договорів та реклами. Наприклад, є обов'язковим вказання певних значень, таких як наприклад річна кредитна ставка, це дозволяє клієнтам простіше оцінювати рівень зобов'язанності при укладенні кредитного договору. [8]

1.2. Проблеми оцінки кандидата на платоспроможність, стан Українського кредитного ринку

Відомо чимало методик та систем оцінки кандидата на отримання позики, адже не існує єдиного правильного рішення, що завжди б давало стовідсоткову точність. Для того, щоб вирізнятися серед конкурентів, а також мати надійну систему, розроблену висококваліфікованими спеціалістами, кожен банк має свою кредитну політику. В ній визначаються власні

скорингові системи, які оцінюють вимоги банку до позичальника, визначаються ризики. Ці системи перевіряються і покращуються роками, базуючись на даних, що постійно поповнюються. А про видачу кредитів на великі суми, зазвичай, окрім програмної оцінки ризиків, збирається кредитна комісія, оцінює та розглядає всі деталі та умови прискіпливіше, веде грунтовну розмову з позичальником.

Умисних шахраїв серед клієнтів кредитних установ небагато. Такий висновок можна зробити проаналізувавши статистичну інформацію, що надається НБУ на офіційному сайті. Визначимо спочатку показник, за яким ми отримаємо таку інформацію - NPL (non-performing loans - частка непрацюючих кредитів).

NPL - показник, що відображає сумарну величину термінової та простроченої заборгованості в кредитному портфелі. Сума враховується у NPL при виконанні хоча б однієї з умов:

1) позичальник банку прострочив виплати на термін, що перевищує 90 днів (30 днів, якщо позика була оформлена у банку-боржнику);
2) позичальник неспроможний виконати зобов'язання по кредитному договору у встановлені строки без процедури стягнення застави.

Тому, чим цей показник менший, тим більше кредитів, що видається, повертається до фінансової структури, приносячи прибуток.

Взагалі, якщо спостерігати за даними, що надає Світовий банк впродовж 2005-2020, то Україна постійно має, порівняно з іншими країнами, дуже високу частку NPL у відношенні до загальної суми валових позик.
Насправді, 20% неповернених кредитів це вже досить немаленька сума. [9]

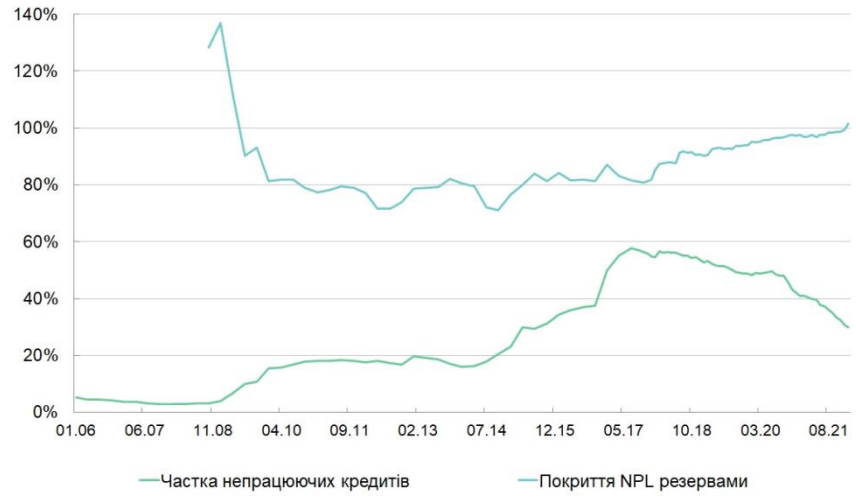


Рисунок 1.1 NPL та резерви фінансових структур України за даними НБУ

Дані, наведені на рисунку 1.1, НБУ пояснює так: «Висока частка NPL - результат кредитної експансії минулих років, коли стандарти оцінювання платоспроможності позичальників були низькими, а права кредиторів недостатньо захищеними. Інша вагома причина – практика кредитування пов'язаних осіб, що припинили обслуговувати кредити під час кризи». [10] Окрім того, за графіком NPL можна помітити часові точки, коли відбулися суттєві інфляції гривні та, відповідно, подорожчання виплат по кредитах: початок 2009, 2014.

Видно, що навіть попри те, що NPL все ще на досить суттєвому рівні, резерви весь час повністю перекривають не виплачені кредити, тож вони не є загрозою до банкрутства і не тиснуть на прибутковість банків загалом. Окрім того, можна помітити, що з 2018 року частка NPL стабільно зменшується, що означає, що банки поступово почали розбиратися з не закритими кредитами, списуючи, реструктуруючи та продаючи борги, що не можуть бути виплачені зараз.

На початок 2022 року частка NPL для приватних банків та банків з іноземним капіталом, без врахування банків Російської Федерації, впала нижче 10%. Основна частина, яка навантажена NPL - державні установи, на

частку яких припадає понад 70% від усіх проблемних кредитів, хоча їх ситуація теж поступово покращується. [10]

На стадії прийняття рішення про підписання договору, щодо надання кредиту, без врахування кризових ситуацій чи проблем із законами для фінансових установ цієї сфери, фінансово-кредитні установи мають можливість досить ефективно визначати платоспроможність позичальника.

Безпомилкова точність неможлива без перенавчання чи у ситуаціях, коли задача залежить від людей та їх можливостей. Тому, свідоме чи неможливе повернення кредиту, що є FN - false-negative (помилки другого роду), звичайно, буде. Завдання аналітиків, що працюють у цьому напрямку, полягає в тому, щоб мінімізувати частоту FN випадків. Окрім того, задача аналітиків також полягає мінімізації частоти помилок FP - false-positive (помилки першого типу). Помилка FP для даної предметної області - це ситуація втрати потенційного прибутку, коли клієнт в кінцевому випадку міг би повернути кредит, але рішення на кредитування не було затверджено і договір не було підписано. Зазвичай FN помилки мають більший вплив на навчання, ніж FP. Для задач встановлюється матриця помилок, за допомогою якої визначається як сильно штрафувати модель за прийняті помилкові рішення.

1.3. Основні впливові чинники на прийняття рішення про видачу кредиту

Законом України про споживче кредитування сформовано основний перелік інформації, що має бути надана кредитодавцю. Зокрема, ціль та мету отримання кредиту, строки надання та терміни і суми виплат по них, рівень доходу людини, що бажає оформити кредит, та іншої інформації та згод на

вимогу. [8] Кожна кредитна установа має власний перелік інформації, що має бути отримана, крім основної частини, зазначеної законодавством. Навіть у стандартизованому шаблоні договору, за цим Законом, також вказано, що можуть вимагатися до надання дані в залежності від системи кредитоспроможності оцінки споживача конкретної фінансової установи, наприклад, майнове та сімейне положення позичальника і багато іншої персональної інформації загалом. Окрім інформації, що надається самим споживачем, за необхідності банк може використовувати у своїх оцінках інформацію отриману з інших законних джерел. Наприклад, перевіряти вже отриману від клієнта інформацію на правдивість. Важливо, щоб надана інформація відповідала дійсності, як і усі надані документи, інакше договір не буде підписано. Якщо при підписанні договору клієнт користується послугами кредитного брокера, то у такому випадку саме він несе відповідальність за достовірність та повноту зібраної та наданої до кредитної установи інформації.

Жодна установа по видачі кредиту не розповсюджує методи за якими проводить оцінку потенційного позичальника, адже використовує різну інформацію із різним впливом на скоринговий бал, але якщо відмова у кредиті відбулася за якихось суттєвих причин, клієнту зазвичай повідомляють. Розглянемо більшість подібних ситуацій.

Найзвичніші відмови на запити про видачу кредиту стосуються ненадання усієї інформації про себе та свій фінансовий стан, документів та згод самим клієнтом. Неповнота інформації - чинник, що може суттєво вплинути на внутрішню оцінку ризиків, чи призвести до проблем на етапах обробки даних. Кредитна установа не має права оформлювати договір за відсутності навіть згоди на обробку та використання персональної інформації. За порушення прав споживачів, наприклад передачу інформації третім особам, кредитор несе відповідальність, крім передбачених законом

випадків.

Наступний суттєвий чинник, що береться до уваги банком - кредитна історія. На надання доступу до інформації якої, клієнту теж обов’язково треба підписати згоду, без якої договір не може бути укладено. Збір та зберігання, зміна та поширення інформації, що стосується клієнта, усіх кредитів, в яких він приймав участь, та порядку їх погашення, відбувається через одне з бюро кредитних історій офіційно включених до Єдиного реєстру бюро кредитних історій. На початок 2022 року таких бюро в Україні налічується 7. [11,12]

Якщо клієнт не має кредитної історії, що є нормальною ситуацією при оформленні першого кредиту після досягнення 18-ти річного віку, після переїзду до України чи невеликого часу після першого кредиту, коли його ще не додано до бази. Треба враховувати, що пересилання інформації про користувача та стан кредиту до бюро відбувається за домовленим з фінансово-кредитною установою графіком. До клієнтів із відсутньою кредитною історією будуть завищені пороги надання кредиту, адже надання позики без відомої історії клієнта - завжди певною мірою ризик. Цікаво, що, при оформленні картки юніора в Приватбанку до літа 2019 року дітям з 6 років (за дозволом та порукою батьків) була доступна функція обмеженого мікрокредиту, тож банк зберігає у себе і такого роду історію.

За статистикою Українського бюро кредитних історій вісімдесят відсотків відмов у видачі кредиту відбувається через негативну кредитну історію клієнта. На офіційних сайтах бюро, а також, зазвичай, у банках, можна перевірити власну кредитну історію в будь який момент. Але за тим же Законом про споживче кредитування, якщо відмову у наданні кредиту будо надано за причини поганої кредитної історії, клієнту безоплатно може бути надана відповідна інформація. [8] Виправити негативну кредитну історію складно, адже переважна більшість кредитних установ не візьмуть на

себе серйозних ризиків неповернення позики. У кожного бюро власні положення, щодо зберігання історії, зазвичай вона консервує історію до 10-річної давності з терміну закриття справи про кредит. Деякі кредитні установи все ж можуть і вдатися до надання ризикових кредитів, але вищі ризики, зазвичай, тягнуть за собою підвищені кредитні ставки. Урахування кредитної історії в будь-якому разі буде відбуватися. Досить сильно вплинути на прийняття подібного рішення можуть особливості ситуації та поважні причини затримок за попередніми платежами, давність кредитної історії. Оформлення кредиту на невелику суму і короткий термін також є фактором який суттєво впливає на довіру до позичальника. Окрім того, іноді банки вдаються до послуг інших кредиторів із метою страхування кредиту. Таким чином, оцінка усіх факторів вже відбувається не однією установою, а комплексно.

Клієнти банків з позитивною кредитною історією, особливо в межах одного банку, зазвичай, можуть мати більші ліміти позик чи строки їх повернення.

Стосовно фінансового стану як фактору, що впливає на рішення про підписання кредитного договору - це ще один дуже важливий показник. Як казалося вище, важливо, щоб уся інформація була достовірною та відкритою. Приховування активних позик у інших банках - поганий знак для кредитної установи. Неофіційні джерела заробітку не враховуються, адже про них не можна отримати та подати довідку. Навіть за гарної кредитної історії, мінімальні шанси, що фінансово-кредитна установа видасть позику, якщо очевидно, що людині доведеться сильно економити на житті. За таких випадків, краще розглянути кредитні умови із більшими термінами.

Ціль кредиту, вік, стать, освіта, посада, стаж роботи, сімейне та майнове положення, навіть активний статус у певному банку може вплинути на рішення про підписання кредитного договору. У деяких банках іноді

безкоштовно, іноді за певну суму, можна отримати власну скорингову оцінку потенційного позичальника та шкалу до неї із групами ризику за певними порогами, але що саме вплинуло на рейтинг дізнатися не вийде.

Прикладом різносторонньої оцінки послугує купівля квартири у кредит. Зазвичай, подібні великі витрати розбивають на великі терміни аж до 20-30 років. За такий проміжок часу багато чого може трапитися, саме тому банки потребують детальних знань про клієнта, адже, наприклад клієнт з певним рівнем освіти зможе з більшою ймовірністю врешті-решт закрити кредит. Цікаво також, що у багатьох банків є верхня межа за віком на підписання договору на позику, вона становить 65-70 років, такого роду інформація зазвичай відкрита.

Окрім того, НБУ періодично проводить опитування банків, небанківських установ та українців стосовно їх кредитного та фінансового становища загалом, тож певні статистичні дані ми можемо розглянути. Опитування, що мало досить цікаві питання серед громадян і буде розглянуто нижче, відбулося 2017 року. [13]

Рис.1.2 Причини відкриття кредиту, % респондентів (1260 осіб)

Зрозуміло, що кредити на лікування, купівлю чи ремонт житла, купівлю машини, кредит на початок власного бізнесу чи освіту - усе кредити, що потребують значних сум. Але навіть за такого випадку, переважна більшість кредитів на суму до 5000 грн - позики на щоденні витрати - так само як і переважна більшість заборгованостей на термін більше 90 днів.

Окрім того, 79% респондентів (серед 2410 осіб) вказали, що не отримували відмови у наданні кредиту за останній рік. Це можна вважати за гарні системи оцінки кредитоспроможності у кредитних установах, адже, відповідно до рисунку 1.4, відсоток затримок із платежами за споживчими кредитами досить низький навіть із зосередженням кредитних сум у досить великих значеннях - рисунок 1.3. Відповідно до інших статистичних оцінок, суттєвого гендерного впливу на рішення про підписання договору не помічено, окрім того, що самотні жінки, що мають на утриманні інших осіб, частіше стикаються з відмовами.

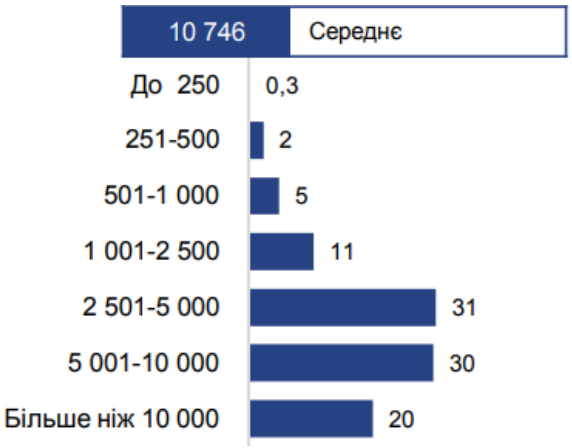


Рис.1.3 Видача суми кредиту за останній рік (в середньому за 1 кредит на 1 респондента), грн, % респондентів (1260 осіб)

Рис.1.4 Затримки платежів респондентами за кредитом, взятим за останній рік, % респондентів (1260 осіб)

1.4. Сучасний стан рішення подібних задач та огляд існуючих рішень

Як вже було згадано раніше, детальні програмні продукти, які наразі використовуються кредитними установ, відсутні у відкритому доступі. Фінансово-кредитні установи при веденні управління та оцінок ризиків мають справу з багатьма видами ризиків, що впливають на всю галузь загалом: кредитний ризик, ринковий ризик, валютний ризик, ризик незбалансованості ліквідності, відсотковий ризик, операційний ризик. [14]

Існує багато методів управління та запобігання різним типам ризиків, при видачі кредитів, посилена увага зосереджується на великих ризиках, пожорщуються кредитні ставки для певних груп населення, чи за певних умов кредитування. Розглянемо етапи оцінки кредитного ризику. Перший етап стосується аналізу даних попередніх років, кожна кредитна установа проводить подібну оцінку у себе із установленою періодичністю, НБУ регулярно статистично зводить та аналізує зібрану інформацію у Звітах про фінансову стабільність (двічі на рік) та Оглядах фінансового сектору (кожного місяця). [15] Тож використовуючи регулярні дані НБУ (можна робити висновки про стан кредитного ризику на рівні країни та враховувати

певні особливості його впливу, застосовуючи методи управління ризиками. Другий етап - етап оцінки наявних даних за конкретною заявкою із урахуванням попередньої інформації.

Наразі найпоширеніший метод оцінки індивідуального кредитного ризику в українських фінансово-кредитних установах це скорингові моделі. [16] Від клієнта отримується інформація за заявкою, далі у відповідності до кожного показника виставляються бали за певними внутрішніми нормативами. Для ухвалення рішення про погодження кредитного договору анкета позичальника має набрати стільки балів, щоб перевищити певну межу, яка відповідає за беззбитковість кредиту. Важливо зазначити, що для гарної роботи цього методу, він має бути не скопійованим, а бути розробкою конкретного банка відповідно до специфіки його клієнтської бази, особливостей управління, аналізу та використанню інформації, що стосується управління ризиками у конкретному регіоні.

Позитивними сторонами скорингового методу є:

1) врахування індивідуальності кожного позичальника та його заявки, адже дозволяє повноцінно врахувати усі параметри відповідно до цілей клієнта та його особливостей, ніякі не пропустивши та не обезцінивши;
2) мінімізація суб’єктивності оцінки, адже вона ведеться із використанням попередньо заданих аналітиками таблиць;
3) легкість програмної реалізації.

Негативними сторонами цього методу є:

1) не зважаючи на популярність та поширеність методу в Україні, досвід інших країн у застосуванні цього методу досить складно адаптувати під українські умови. Велика кількість відмінностей, які були запроваджені досить давно, наприклад, перерахування віку позичальника у негативні бали, у багатьох європейських

країнах враховуються навпаки. Іншим прикладом може слугувати показник кількості та частоти зміни місця працевлаштування, який у Сполучених Штатах вважається позитивним фактором, коли в Україні за таким показником бали віднімаються. Саме тому, українські скорингові моделі можуть мати проблеми із точністю проведення оцінок;

2) система, що проводить скорингову оцінку, потребує систематичного оновлевлення та доопрацьовувань, через короткостроковість системи роздачі балів. Це лягає важким тягарем на плечі тих аналітиків, які, необхідно, щоб мали великий досвід у відповідальній частині перерозподілу системи розстановки балів;
3) через специфіку заробітків в Україні, скорингова модель має проблеми із оцінюванням реального офіційного доходу, адже виставлення суттєвих балів джерелам неофіційного доходу - не є надійним рішенням.

Взагалі, скорингова модель використовується не тільки для оцінки кредитоспроможності позичальника, а має багато типів, для використання у різних задачах банківської діяльності, навіть на одному напрямку, що стосується видачі кредиту:

Таблиця 1.1 Типи скорингових систем [16]

#	Тип скорингу	Характеристика
1.	Скоринг заявника	Типовий скоринг для оцінки кредитоспроможності позичальника при отриманні кредиту.
2.	Колекторський скоринг	Скоринг, спрямований на вирішення питання проблемних кредитів банку до моменту їх передачі колекторам.
3.	Поведінковий	Спрямований на прогнозування кредито- та

	скоринг	платоспроможності позичальника на основі особливостей його поведінки в минулому. Дає можливість оцінити динаміку стану кредитного рахунку клієнта.
4.	Скоринг проти шахрайства	Скоринг, за допомогою якого аналізується можливість шахрайства з боку позичальника.
5.	Скоринг реакції	Спрямований на оцінку реакції позичальників щодо послуг, що пропонуються банком для них.
6.	Скоринг втрат	Оцінює ймовірність користування кредитним продуктом банку в зв'язку із зміною умов функціонування.
7.	Передпродажний скоринг	Скоринг для генерації попередньо затверджених (preapproved) пропозицій клієнтам.

Оцінка кредитного ризику для кожної заявки - індивідуальна справа, , що потребує унікального підходу, сильно залежить від досвіду роботи, якості та навичок кредитного менеджера, особливостей ситуації в країні та багато іншого. Українське бюро кредитних історій розміщує на своєму сайті, наприклад, пропозицію побудови індивідуальної скорингової моделі заявника на власних даних із точністю 79% [17]. Застосування інтелектуальних аналітичних методів при управлінні кредитними ризиками може суттєво вплинути на якість скорингових моделей, частково прибравши недоліки. Існує багато моделей оцінки кредитоспроможності позичальника:

Рис.1.5 Моделі оцінки кредитного ризику та кредитоспроможності позичальника [16]

Але взагалі, треба пам'ятати, що задача оцінки кредитоспроможності позичальника, навіть із проведенням скорингу, має у меті вирішення задачі бінарної класифікації, тобто фінальний бал неважливий, якщо вже відомо чи збиткова видача певного кредиту, чи ні. Вплинути на показник кредитного ризику можна наприклад із застосуванням дерев класифікації, розподіляючи клієнтів у групи ризику і формуючи унікальні скорингові шкали. Для оцінки платоспроможності ніхто не забороняє використовувати такі ж методи для бінарної класифікації, як і у не таких серйозних галузях. Класифікація - найбільш поширена задача аналітики даних, тим паче бінарна класифікація, до якої можуть бути зведені складніші задачі. Тому методів для автоматизованого вирішення дуже багато:

Для автоматизованого вирішення задач бінарної класифікації часто застосовують методи:

1) Decision Tree (Дерево рішень)
2) Random Forest (Випадковий ліс)
3) Logistic Regression (Логістична регресія)
4) SVM - support vector machine (Метод опорних векторів)
5) Naive Bayes (Наївний баєсів класифікатор)
6) k-Nearest neighbors - K-NN (K-найближчого сусіда)
7) Штучні нейронні мережі

Не існує ідеального методу для усіх задач, у кожної задачі свої особливості, специфіка подачі даних, змінні параметри.

1.5.Постановка задачі

Метою цього дипломного проекту створення інтелектуальної інформаційної системи, яка проводитиме скорингову оцінку платоспроможності позичальника за отриманими від нього даними та, після вирішення задачі класифікації, надаватиме рекомендації користувачу веб-застосунку як фінансово-кредитній установі про погодження чи відмову на підписання кредитного договору.

Об’єкт дослідження: надання рекомендацій для якнайменш збиткового прийняття рішення щодо видачі позики

Предмет дослідження: методи та засоби автоматизованого визначення кредитоспроможності позичальника.

Основні зацікавлені сторони у задачі оцінки кредитоспроможності позичальника та результатах отримання якнайточніших оцінок:

1) банки/небанківські фінансові установи
2) представники Fintech
3) позичальники

Рис.1.6 контекстна діаграма системи інтелектуального застосунку оцінки кредитоспроможності позичальника у нотації IDEF0

Таким чином, можна виділити основні етапи проведення аналітики оцінки кредитоспроможності позичальника, які наведені на рис. 1.7.

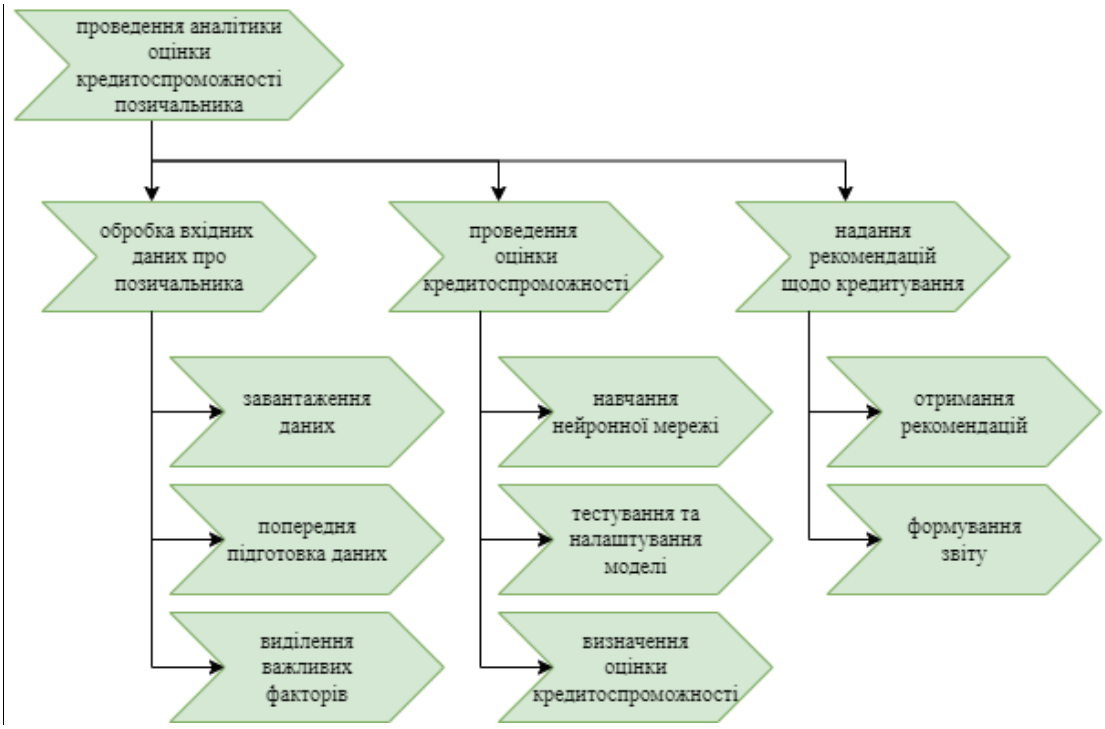


Рис.1.7 основні етапи при проведенні аналітики оцінки кредитоспроможності позичальника у нотації VAD

Вхідні дані суттєво залежать від нормативних документів та розробок аналітиків фінансово-кредитної установи, що стосуються факторів оцінки кредитоспроможності, тому в залежності від даних, перелік факторів може

відрізнятися від наведеного нижче:

Таблиця 1.2 Класифікація факторів [18]

Соціально економічні	Фінансові	Професійні	Особисті
місцезнаходження	прибуток від основного місця роботи	поточне працевлаштування	значення проведеного скорингу
вік	додаткові джерела прибутку	категорія компанії поточного місця роботи	співвідношення ліквідності активів і зобов'язань
досвід роботи	перспективні джерела прибутку	професійні навички	сімейний статус
профіль соціальної взаємодії	обрана модель кредитування		міжособистісні стосунки

Вихідними даними є оцінена можливість позичальника повернути кредит за вказаними умовами та рекомендація від системи щодо релевантності підписання кредитного договору з ним.

Вимоги до створюваного програмного застосунку:

Функціональні вимоги:

1) З користувацької сторони мова застосунку має бути українською або англійською.
2) Має бути доступною можливість зчитування навчальних та тренувальних даних у табличних форматах. (.csv)
3) За допомогою підготовчої частини розробки, аналітиком має бути виконана попередня обробка даних, налаштована та навчена модель;
4) Після навчання найкраща модель має бути збережена для

уникнення зайвих витрат часу;

5) Має бути реалізований зручний користувацький інтерфейс, що, на обробному етапі, дозволить заповнити поля про клієнта демонстраційними даними;
6) Має проводитися оцінка платоспроможності клієнт, формуватися рекомендації щодо видачі кредиту;
7) За бажанням користувача має бути збережено анкету та рекомендації щодо клієнта у файловому форматі.

Нефункціональні вимоги:

1) Точність результатів має бути не меншою за 70%;
2) Етап навчання нейронної мережі має займати до десяти хвилин;
3) Обробка одиничних запитів має відбуватися швидше ніж за 3 хвилини;
4) Користувацький інтерфейс має бути інтуїтивно зрозумілий та привабливий;
5) Поля введення даних мають бути захищені від вводу даних некоректного типу;
6) Система має забезпечувати надійну роботу, гарантувати відсутність програмних збоїв;
7) Застосунок має мати зрозумілу файлову структуру збереження.
8) Система має бути розроблена відповідно до норм діючого законодавства України.

Передбачається, що договір про обробку персональних даних користувача підписується поза межами веб-застосунку.

1.6. Висновки до розділу 1

Розділ передбачає детальний аналіз предметної області, а саме, області надання кредитних послуг, формування теоретичної бази та постановку задачі для подальшої роботи.

Було розглянуто методики та проблеми оцінки кредитоспроможності позичальника, а також, вплив конкуренції між фінансово-кредитними установами. Описані існуючі рішення та сформовано базову постановку задачі. Описано функціональні та не функціональні вимоги.

РОЗДІЛ 2 ПРОЕКТНІ РІШЕННЯ ТА ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1. Структура набору даних

Обраний набір даних містить 1000 записів з 20 категоричними/символічними атрибутами, підготовленими професором Гофманом з інституту статистики з університету економіки в Гамбурзі (Німеччина) 17 листопада 1994 року. У цьому наборі даних кожен запис представляє особу, яка бере кредит у банку. Кожна людина класифікується як хороший чи поганий кредитний ризик відповідно до набору ознак. Останній, двадцять перший стовпчик містить прийняті банком рішення, щодо видачі кредиту (1 = добре, 2 = погано). [19]

Атрибути набору даних для детального ознайомлення наведено у таблицях у ДОДАТКУ А. (я - якісний атрибут; к - кількісний атрибут)

Пропущені значення та рядки, що повторюються, відсутні. На рисунку 2.1 наведено приклад того, як виглядають дані у необробленому вигляді.

	col_1	col_2	col_3	col_4	col_5	col_6	col_7	col_8	col_9	col_10	col_11
0	A11	6	A34	A43	1169	A65	A75	4	A93	A101	4
1	A12	48	A32	A43	5951	A61	A73	2	A92	A101	2
2	A14	12	A34	A46	2096	A61	A74	2	A93	A101	3
3	A11	42	A32	A42	7882	A61	A74	2	A93	A103	4
4	A11	24	A33	A40	4870	A61	A73	3	A93	A101	4

	col_12	col_13	col_14	col_15	col_16	col_17	col_18	col_19	col_20	y
0	A121	67	A143	A152	2	A173	1	A192	A201	1
1	A121	22	A143	A152	1	A173	1	A191	A201	2
2	A121	49	A143	A152	1	A172	2	A191	A201	1
3	A122	45	A143	A153	1	A173	2	A191	A201	1
4	A124	53	A143	A153	2	A173	2	A191	A201	2

Рис.2.1 Перші п’ять елементів необробленого набору даних

Проведення класифікації на задачі такої великої розмірності, насправді досить складна задача, яка вимагає вдалого підбору параметрів моделей та якісного проведення попередньої обробки даних. Іноді допомогти із

розумінням даних може допомогти проектування їх у двовимірний простір. Але у даному випадку з рисунка 2.2 можна зрозуміти, що про ніяку візуальну роздільність мова не буде заходити.

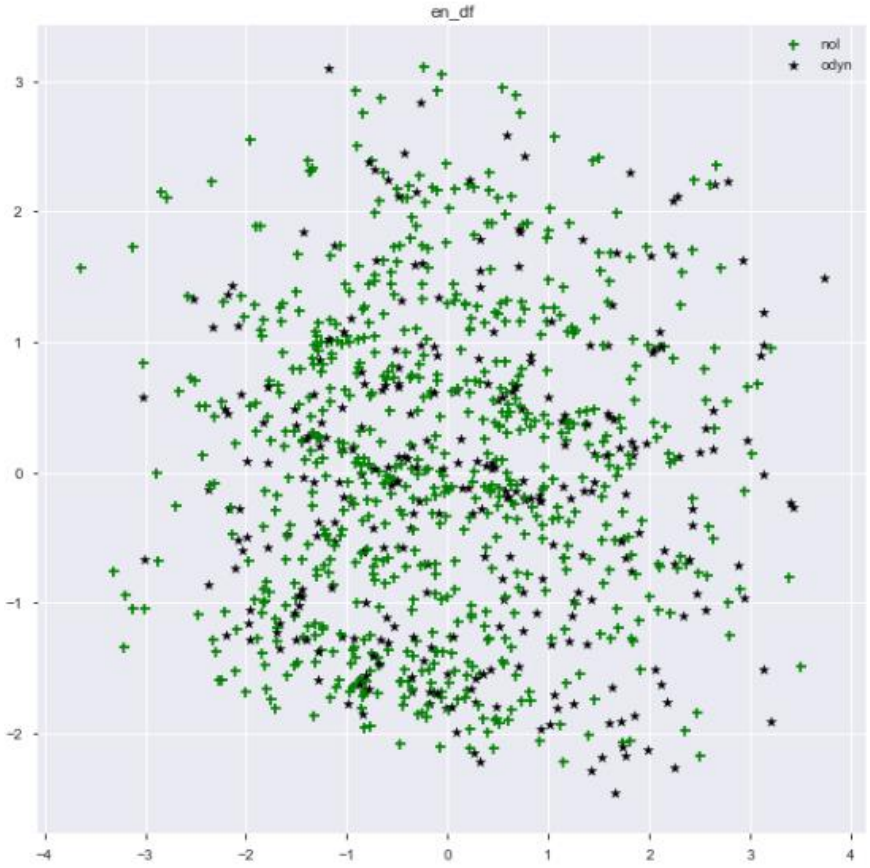


Рис 2.2 Проекція даних у двовимірний простір (зелені плюсики - люди, що повернули кредит, чорні зірочки - не повернули)

2.2. Короткі теоретичні відомості про методи аналізу даних

Предмет вивчення в даній роботі фактично методи вирішення задачі класифікації. Задача класифікації - розбиття множини спостережень на певні попередньо задані класи. У межах класу об’єкти мають достатньо схожі властивості та ознаки, щоб бути згрупованими. Бінарна класифікація - найпростіший випадок, до якого за можливості намагаються зводити більш

складні задачі. Наприклад, замість визначення таких ступенів кредитного ризику, як «Високий», «Середній» або «Низький», можна використовувати всього два - «Видати» або «Відмовити». [18]

Процес рішення задачі класифікації складається із двох етапів - формування моделі та застосування моделі. Формування моделі здійснюється за допомогою навчання на навчальній вибірці. Навчену модель можна представити у вигляді певних правил, наприклад у вигляді дерева у найнаочнішому варіанті, які по суті є набором математичних порогових правил Навчену модель необхідно оцінити, бажано декілька разів, наприклад використовуючи методику крос-валідації - класифікація нових даних має сенс виконуватися тільки після отримання точності моделі. [20]

Основні проблеми, з якими зустрічаються аналітики даних у процесі знаходження розв’язку та налаштуванні моделі для вирішення задачі класифікації

1) неякісні дані (дублювання спостережень, пропущені значення, викиди та інші);
2) взаємопов’язанність даних;
3) різна важливість атрибутів та відмінності розподілів у них;
4) overfitting і underfltting.

На етапі попередньої обробки даних проводиться відсіювання аномальних значень, заповнення апроксимацією чи вилучення рядків із пропущеними комірками, перетворення якісних атрибутів у зручні для подальшої роботи.

Зазвичай висока кореляція між атрибутами не дуже корисна, іноді заради зменшення розмірності задачі сильно корельовано стовпчики видаляють. Окрім того видалення атрибутів проводиться, якщо вони мають незначну кореляцію із цільовим стовпцем. Такі зміни допомагають не тільки пришвидшити навчання моделі, а і покращити її якість за рахунок того, що

зменшується кількість атрибутів, на які модель буде «відволікатися», розсіюючи точність.

Усі вхідні змінні бажано привести до єдиного діапазону та нормувати, адже після кодування інформації - отримуються різнорідні величини, що змінюються у різних діапазонах. В іншому випадку помилки, зумовлені змінними, що змінюються в широкому діапазоні, будуть впливати сильніше, ніж помилки змінних, що змінюються у вузькому діапазоні. [21] Кожна вхідна змінна масштабується незалежно від інших змінних.

Суть overfitting (перенавчання) полягає в тому, що модель із часом занадто добре адаптується до даних, тобто описує помилкові дані як частину правил набору даних. Така модель потенційно може мати не таку і низьку точність, але у більшості випадків точність падає із наростанням перенавчання, адже тестові дані можуть виходити та вузько окреслені перенавченою моделлю рамки. Відслідкувати перенавчання можна порівнявши точність на навчальній вибірці із точністю на валідаційній. На валідаційній вона менша.

Зворотній випадок underfltting (недонавчання) - коли точність на навчальній вибірці занадто нижче від очікуваної.

У обох випадках варто передивитися підхід до виявлення закономірностей у даних. Зменшити-збільшити кількість ітерацій, переналаштувати параметри до того моменту, поки модель не зрівняє точності валідаційної вибірки та навчальної.

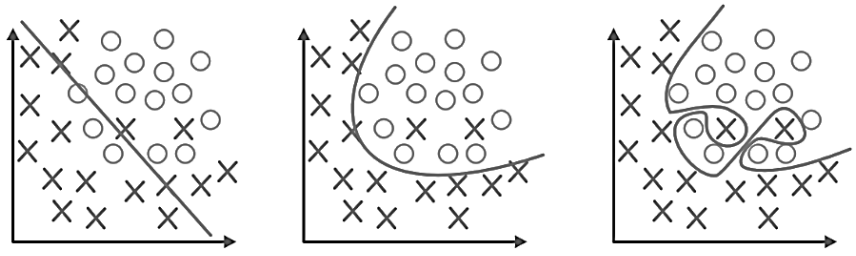


Рис 2.3 Недо-, нормальне, пере- навчання (зліва направо)

Навчання з учителем є одним із основних напрямків у інтелектуальному аналізі даних у класі задач класифікації. Воно дозволяє достатньо швидко зменшувати помилки точності класифікації за рахунок коригування вагових коефіцієнтів із виростанням порівняння отриманого значення із істинним значенням цільової функції. В певному роді навчання з учителем можна визначити як метаевристику для пошуку найкращих правил розділів класів. [22]

2.2.1. Штучна нейронна мережа

Це широкий напрямок систем для вирішення великої кількості різнотипних задач машинного навчання. Нейронна мережа імітує роботу нейронів у мозку, шляхом побудови ієрархічної мережі, в якій нейрони вищих рівнів поєднані з входами нейронів нижчих рівнів.

Розглядатимемо повнозв’язну нейронну мережу прямого поширення (FNN - feedforward neural network). Вона є одним із найпопулярніших і найбільш широко використовуваних видів моделей для вирішення багатьох практичних задач. Ще одна назва такої архітектури нейронної мережі - multilayer perceptron. Архітектуру цієї мережі наведено на рисунку 2.4, у неї кожен нейрон попереднього шару з’єднаний із кожним нейроном наступного шару.

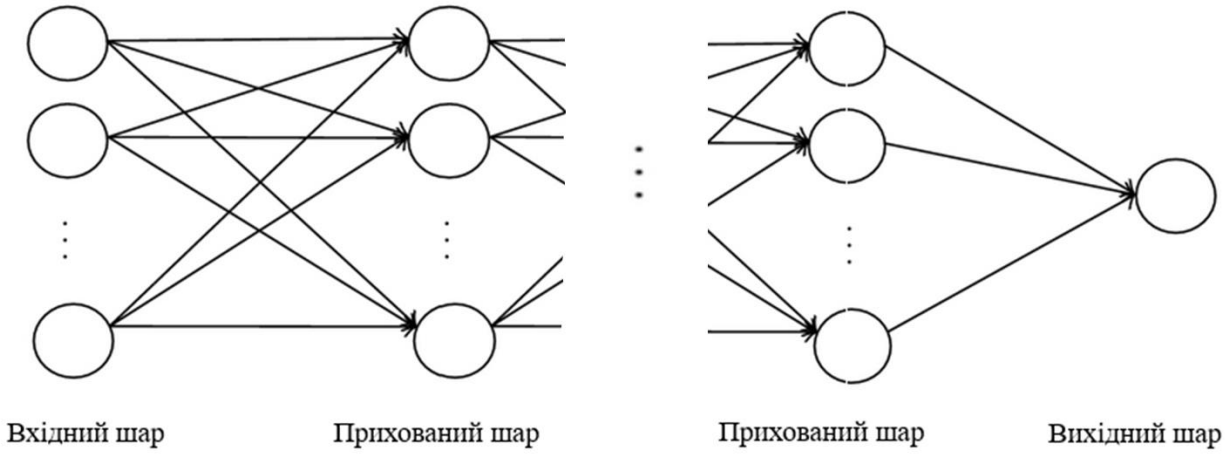


Рис 2.4 Архітектура нейронної мережі прямого поширення

На нейрони вхідного шару (зліва) подаються значення вхідних параметрів, на основі яких потрібно приймати рішення про належність до класу, прогнозувати значення тощо. Ці значення передаються у наступний шар, коригуючись відповідно вагових коефіцієнтів, що визначаються міжнейроними зв'язками. У результаті на вихідному формується значення, яке розглядається як реакція усієї мережі на отримані значення вхідних параметрів. Немає сенсу запускати мережу прямого поширення тільки на одну ітерацію. Важливим елементом алгоритму навчання мережі є методика зворотного поширення помилки (back propagation), вона базується на δ-правилі. Після завершення ітерації, тобто прогону вхідних значень до вихідного шару, знаходиться похибка між тренувальним значенням вихідної функції та отриманим нейронною мережею. Відповідно до цієї похибки проводиться корекція ваг по кожному міжнейронному зв’язку. [23] Подальше тренування полягає у ітераційному підборі коригуванням ваг міжнейроних зв'язків, до тої міри, поки похибка не буде мінімізована якнайкраще.

Перевагами нейронних мереж є висока точність прогнозування, а недоліком - складність її налаштування, велика тривалість навчання, а також є необхідність у дуже великому обсязі навчальної вибірки.

Ще один істотний недолік такий: натренована нейронна мережа - це

«чорна скринька». Знання, зафіксовані як ваги кількох сотень міжнейронних зв’язків, людина не в змозі проаналізувати й інтерпретувати. [24]

2.2.2. Методи вилучення викидів

Метод IQR interquartile range (міжквартильний розмах) -, це метод, що використовується для відсічення викидів. Для очистки наборів даних із нормальними розподілами рекомендується використовувати інший метод - метод трьох сігм, іншими словами - метод трьох середньоквадратичних відхилень.

Візуально доступний приклад застосування методу IQR наведено на рисунку 2.5.

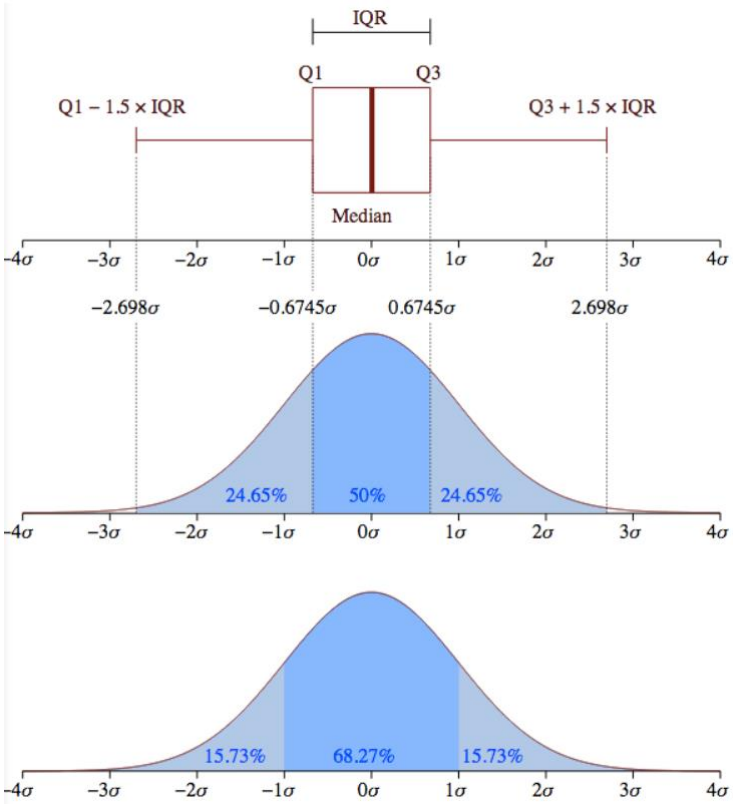


Рис 2.5 Приклад застосування методу IQR у порівнянні із методом трьох сігм.(де Q1 та Q3 - 25-й і 75-й процентілі)

Для зручної інтерпретації методу прийнято використовувати для його пояснення діаграму ящика із вусами (boxplot). Сам IQR це фактично відрізок між 25-м і 75-м процентілями набору даних, у якому знаходиться рівно половина значень набору даних. Метод IQR вважає за викиди те, що лежить поза 25-м процентілем із IQR та ще його половинкою зліва, а також те, що лежить поза 75-м процентілем із IQR та ще його половинкою справа. Тобто викидами вважаються усі точки, що знаходяться поза проміжком [Q1-1.5*IQR , Q3+1.5*IQR].

Метод трьох сігм, що використовується для нормально розподілених даних, описує викидами все, що лежить поза трьома стандартними відхиленнями σ від медіани розподілу в обидві сторони.

Жодний із кількісних атрибутів набору обраних для цієї дипломної роботи даних не має нормальний розподіл, тому використання методу IQR для очистки даних від викидів має сенс бути застосованим.

Методи вибору атрибутів, як і методи прибирання даних-викидів також певним чином відносяться до методів покращення якості та швидкості роботи моделі. Вибір методу вибору атрибутів у загальному випадку залежить від характеру та типу атрибуту. [25] Якщо перед застосуванням методу вибору атрибутів провести декодування даних на одиниці та нулі, і якщо подібна дія взагалі доцільна у відповідності до типу даних, то можна наприклад використовувати звичайний показник Пірсона за міру подібності, якщо є необхідність будувати лінію подібності.

Бінарна класифікація взагалі, як вже згадувалося, спрощена версія типової задачі класифікації, адже в такій задачі, як мінімум можна судити про те, що вихідний стовпчик можна трансформувати у числовий формат. Саме тому будемо використовувати показник подібності, що має усі, і вхідні, і вихідні дані числовими. Показник Пірсона використовується для побудови лінійної залежності між даними. Будемо проводити розрахунки кореляції

стовпців із цільовою ознакою із використанням показника Спірмена, для нелінійних монотонних залежностей. [26]

Тим більше показник кореляції Спірмена не такий чутливий до суттєво віддалених викидів, які могли залишитися невидимими для IQR. Але по суті показник Спірмена це той самий показник Пірсона, але розрахунок якого проведено серед рангів значень даних. Окрім того, на відміну від показника кореляції Пірсона, він не робить припущень про нормальність розподілу даних.

2.2.3. Методи боротьби із незбалансованими вибірками

Серед методів боротьби із незбалансованими вибірками широко застосовуються методи ресемплінгу та методи, базовані на розрахунку ваги класів.

Вагові показники класів безпосередньо змінюють функцію втрат, надаючи більше, або менше покарань для класів з більшою, або меншою вагою. На практиці, метод в основному жертвує деякою здатністю передбачати менш вагомий клас, навмисно зміщуючи модель та надаючи перевагу більш точним прогнозам більш вагомого класу.

Серед методів ресемплінгу (методів роботи із кількістю даних у вибірці) виділяють два основних: генерація нових значень для меншого класу чи видалення значень більшого класу. Методи, які на відміну від зменшення об’ємів даних більшого класу, добре працюють, коли немає великої кількості даних, і скорочувати їх кількість недоцільно. Такі методи обов’язково треба використовувати тільки після етапу розділення даних на тренувальні та тестові, щоб зберегти унікальність тестових даних та забезпечити адекватність проведення оцінки точності. В інакшому випадку може

спричинитися overfitting і в результаті погіршить результати класифікації даних тестувальної вибірки.

Є декілька подібних методів - копіювання випадкових даних цільового класу чи апроксимація із отриманням нових даних за певним методом. Використання обох полягає у створенні штучних рядків. У бібліотеці python imblearn реалізована велика кількість методів ресемплінгу. [27] Наприклад реалізовано SMOTE (Synthetic Minority Oversampling Technique), який базується на виборі місць для генерації нових точок застосовуючи метод найближчого сусіда. Більшість тим чи іншим чином використовують інформацію про сусідні точки для генерації нових. Усі з них мають відмінності у генерації нових точок, використовують різні методи, у певному сенсі розглядають нову точки, як втрачені дані, які необхідно апроксимувати. Порівнювати ці методи візуально досить просто, коли можливо побудувати двохвимірну проекцію даних.

Якщо порівнювати методи ресемплінгу із методом встановлення ваг класів, то методи ресемплінгу також надають більше ваги певним класам - дублювання спостережень дублює покарання за ці конкретні спостереження, надаючи їм більше впливу на відповідність моделі, але через специфіку розподілів даних, результати оцінки якості моделі можуть суттєво відрізнятися.

2.2.4. Засоби оцінки якості моделі

Оцінка якості класифікації зазвичай проводиться за допомогою крос-перевірки. Cross-validation - це процедура оцінки точності класифікації на даних з тестової множини. Береться середнє по результатах (це робиться для того, аби уникнути залежності від вибору спостережень у тестовій вибірці).

Використовується пересічна крос-перевірка так: покроково перебирається поділ на тестові дані та навчальні таким чином, що кожне спостереження потрапляє у тестову вибірку один раз, а у навчальну - (n-1). Важливим параметром при проведенні крос-валідації є на кожній ітерації брати випадкову k-ту частину даних, чи йти по даних послідовно. На рисунку 2.6 можна переглянути приклад проведення крос-валідації без застосування перемішування даних.

	A	B	C	D	E
cross-validation 1 ітерація	тестова вибірка	навчальна вибірка	навчальна вибірка	навчальна вибірка	навчальна вибірка
cross-validation 2 ітерація	навчальна вибірка	тестова вибірка	навчальна вибірка	навчальна вибірка	навчальна вибірка
cross-validation 3 ітерація	навчальна вибірка	навчальна вибірка	тестова вибірка	навчальна вибірка	навчальна вибірка
cross-validation 4 ітерація	навчальна вибірка	навчальна вибірка	навчальна вибірка	тестова вибірка	навчальна вибірка
cross-validation 5 ітерація	навчальна вибірка	навчальна вибірка	навчальна вибірка	навчальна вибірка	тестова вибірка

Рис 2.6 Приклад кросс-валідації при поділі вибірки на п’ять частин (A-E)

На рисунку 2.7 наведено види помилок, що стаються при проведенні будь якої класифікації. На прикладі наведено випадок бінарної класифікації.

	$y = 1$		$y = 0$	
$\hat{y} = 1$	True Positive (TP)		False Positive (FP)	помилка першого роду
$\hat{y} = 0$	False Negative (FN)	помилка другого роду	True Negative (TN)	

Рис 2.7 Матриця помилок, де $\hat{y}$ - відповідь алгоритма, а y - справжня мітка класу на прикладі бінарної класифікації

Рисунок 2.7, та подальший текст вимагають додаткових пояснень для простішого спийняття:

1) TP - true-positive (правильно розпізнано клас 1)

2) TN - true-negative (правильно розпізнано клас 2)
3) FP - false-positive (помилка першого роду)
4) FN - false-negative (помилка другого роду)

Важливим параметром будь-якої моделі для вирішення задачі класифікації є cost-matrix, тобто матриця ціни помилки. Зазвичай помилки першого та другого роду мають різну вагу у реальних задачах. [28]

Accuracy як параметр оцінки якості моделі є відсотком правильних відповідей алгоритму для наборів даних із однаковою кількістю. Простими словами, accuracy це те, що зазвичай розуміється під точністю моделі.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Для набору даних у даній дипломній роботі accuracy не підходить через те, що цільовий стовпчик має нерівномірний розподіл класів цільового стовпчика.

Precision (повнота) і recall (точність) - ортогональні критерії якості, які має сенс покращувати тільки разом:

$$precision = \frac{TP}{TP + FP} \tag{2.2}$$

1) Можна побудувати алгоритм із precision у 100%: він всі об'єкти буде відносити до одного класу, але при цьому показник recall може бути дуже низьким.

$$recall = \frac{TP}{TP + FN} \tag{2.3}$$

2) Можна побудувати алгоритм із recall 100%: він буде відносити до класів тільки ті об'єкти, в яких точно впевнений, при цьому показник precision може бути дуже низьким.

Ці критерії якості стосуються саме конкретного класу, а не усієї моделі, хоча все ще можна для моделі їх розглядати як макро-, мікро-, середню чи середньозважену оцінку.

F1-score, тобто міра F1 - це середнє гармонійне точності і повноти, її максимізація призводить до одночасної максимізації ортогональних критеріїв:

$$F_1 \text{ score} = \frac{2}{\frac{1}{TP/(TP+FP)} + \frac{1}{TP/(TP+FN)}} = \frac{2TP}{2TP+FP+FN} \tag{2.4}$$

Precision і recall, на відміну від accuracy, не залежать від співвідношення класів і тому застосовні в умовах незбалансованих вибірок.

Roc-curve оцінка точності (receiver operating characteristic curve) - ще один критерій, який дозволяє оцінити якість методів класифікації. Він відображає співвідношення між часткою об'єктів від загальної кількості носіїв ознаки, правильно класифікованих (TPR) до загальної кількості об'єктів, що не несуть ознаки, помилково класифікованих, як такі, що мають ознаку (FPR).

$$\text{(True Positives Rate):} \quad TPR = \frac{TP}{TP+FN} \cdot 100\,\% \tag{2.5}$$

$$\text{(False Positives Rate):} \quad FPR = \frac{FP}{TN+FP} \cdot 100\,\% \tag{2.6}$$

ROC-крива також відома як крива похибок. Кількісну інтерпретацію ROC дає показник AUC (area under curve) - площа, обмежена ROC-кривою і віссю частки помилкових позитивних класифікацій. Чим вище показник AUC, тим якісніше діє класифікатор, при цьому значення близьке до 0,5 демонструє непридатність обраного методу класифікації, що, для випадку бінарної класифікації відповідає випадковому вгадуванню.

2.3. Планування та проектування архітектури системи

Однією із поставлених підзадач цього дипломного проекту є необхідність у реалізації інтерфейсу застосунку. Розглянемо можливі варіанти:

1) Використання вбудованих бібліотек мов програмування:

   + повна незалежність від додаткових програм та мороки із їх підключенням;

   - обмеженість функціоналу, складність імплементації дизайнерських рішень інтерфейсу, застарілість базових стилів.

2) Застосунок для персонального комп'ютера:

   + легке тестування, зручність у користуванні, працює без необхідності підключення до мережі.

3) Мобільний застосунок:

   + працює без необхідності підключення до мережі, зручно у користуванні;

   - складності у тестуванні, залежність від потужності телефону.

4) Веб-застосунок:

   + багатокористувацький доступ, легкість у роботі із дизайном та стилями, адаптивність, швидкість роботи, можливість покращення застосунку із різних пристроїв одразу усюди;

   - необхідність доступу до інтернет мережі.

У подібній фінансовій сфері дуже впливовим чинником є можливість ведення певною мірою адміністрування над багатокористувацьким доступом. Реалізація системи ведення логів - серйозний крок який теж треба передбачити. Надавати доступ партнерам, скасовувати його за необхідності, а також зручність подальшого покращення та модифікації застосунку - важливі елементи, адже потенційно розроблюваний програмний продукт

планується у вигляді зручному для його підтримки монетизацією від користувачів.

Великі обчислювальні ресурси для подібної задачі не будуть потребуватися, а у сучасному світі доступність інтернет-мережі не є проблемою без форс-мажорних обставин, окрім того область використання програми не передбачує необхідність його використання у повсякденному житті.

Окрім того важливим побажанням до застосунку вважається привабливість його інтерфейсу. Працювати звісно можливо і на сухих таблицях, але від такого користувач буде досить сильно втомлюватися. Візуально проста та доступна подача отриманих результатів обчислень кредитоспроможності позичальника дозволить не втомлювати корчувача і тим самим спрощувати виконання його задач і складі банківської установи.

Для реалізації поставлених задач було обрано розробку веб-застосунку.

Спроектовано схему архітектури майбутнього веб-додатку:

Рис 2.8 Архітектура веб-додатку для оцінки кредитоспроможності клієнта.

Інтерфейс - клієнтська частина веб-застосунку, через неї відбувається уся взаємодія із серверною частиною - Бекендом. Ця взаємодія відбувається за http протоколом (із використанням POST-GET запитів). Відповідно до запитів користувачів (за умови правильного рівня доступу) витягуються або записуються дані у БД за допомогою ДАО, а також надсилаються дані у модуль оцінки кредитоспроможності позичальника, після чого результати надсилаються назад до інтерфейсу та відображаються у доступному виді.

2.4. Детальний функціональний та процесний аналіз роботи алгоритму

2.4.1. Функціональний аналіз

Опираючись на описані у підрозділі 1.5 функціональні та не функціональні вимоги, створимо конкретне дерево-функцій веб-застосунку (рисунок 2.9).

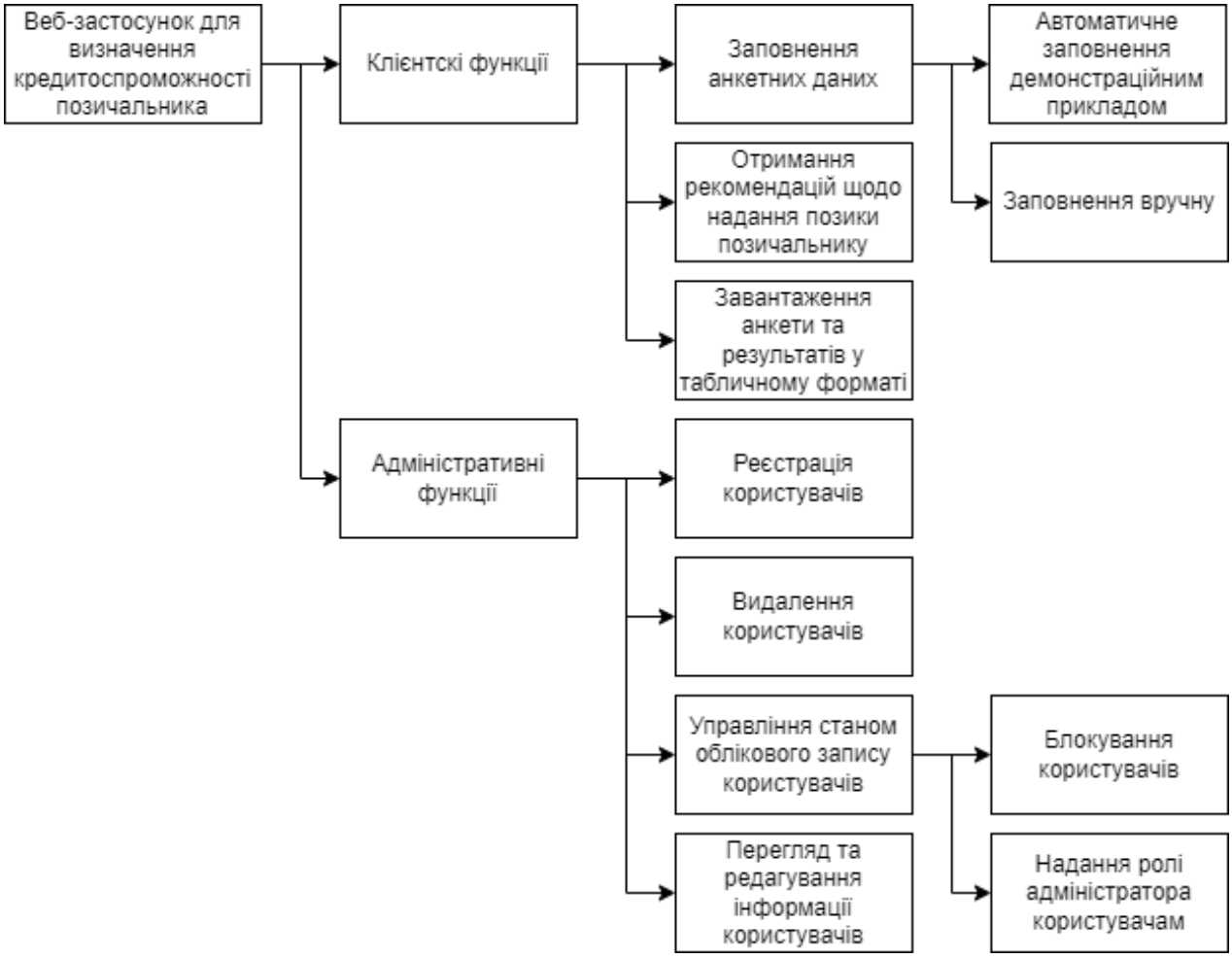


Рис 2.9 Дерево функцій веб-додатку для оцінки кредитоспроможності позичальника

Для повноти розуміння функцій визначимо ролі користувачів веб-застосунку та опишемо їх за допомогою діаграми прецедентів на рисунку 2.10.

Заплановані ролі для користувачів веб-застосунку:

1) Гість - не авторизований користувач, який матиме доступ тільки до сторінки авторизації. (Важливо зазначити, що реєстрація нових користувачів здійснюватиметься контрольовано за участі

адміністратора. Заявку на реєстрацію можна залишити за електронною адресою на сторінці авторизації)

2) Звичайний користувач - має доступ до усього функціоналу веб-додатку окрім адміністративних можливостей
3) Адміністратор - має доступ до усіх функцій.

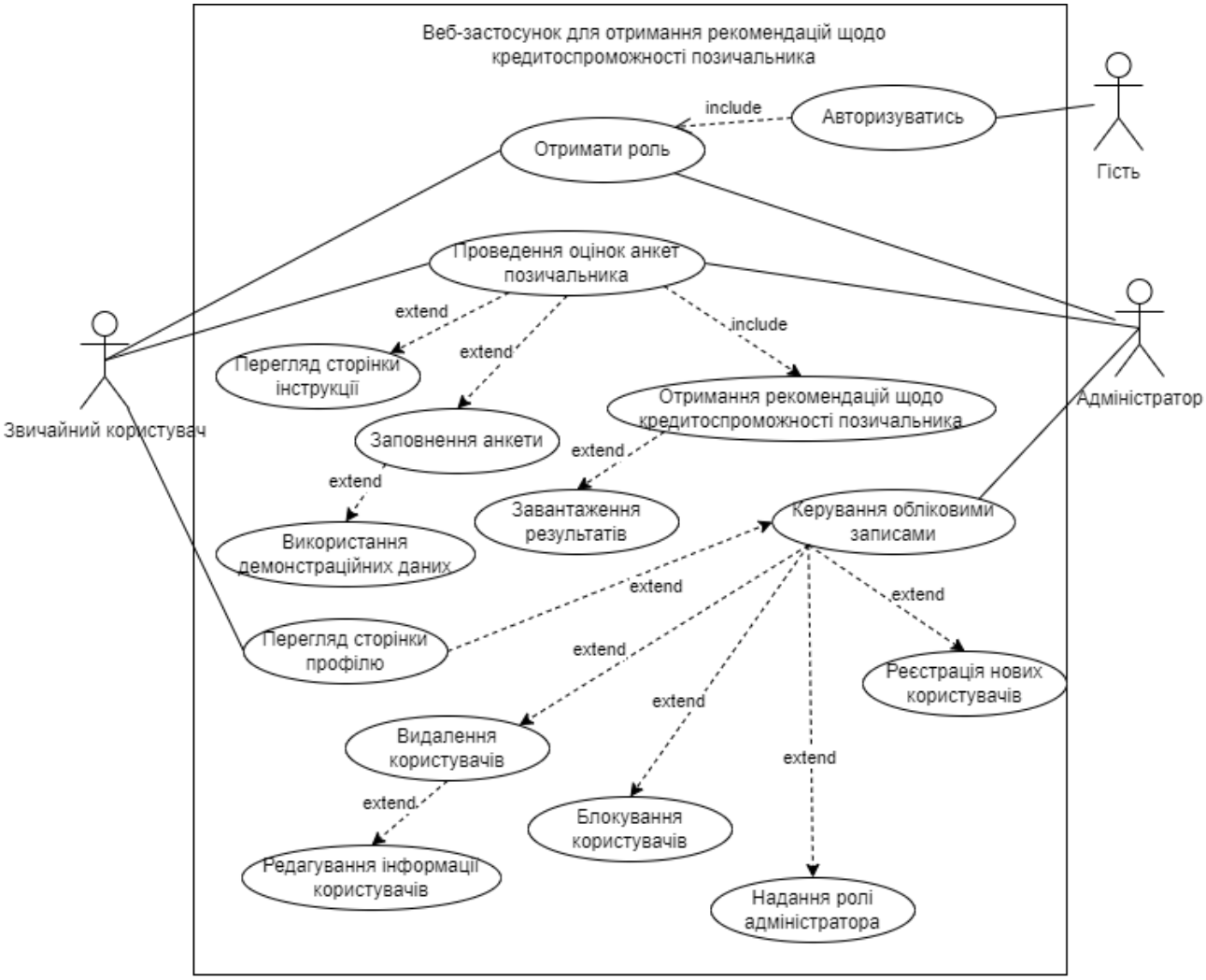


Рис 2.10 Діаграма прецедентів для веб-застосунку для отримання рекомендацій щодо кредитоспроможності позичальника

2.4.2. Процесна деталізація

Проведемо декомпозицію діаграми у нотації IDEF0, наведеної у

підрозділі 1.5 даної дипломної роботи для деталізації (рисунок 2.11)

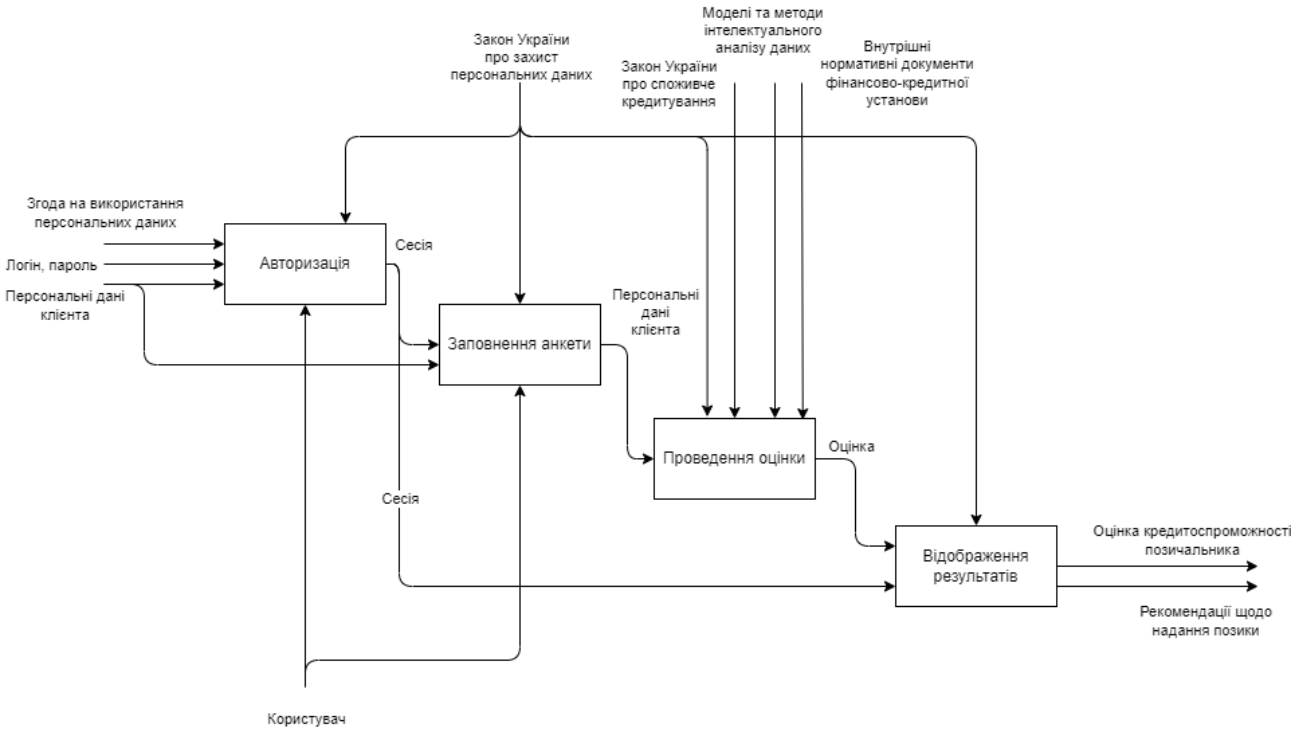


Рис 2.11 Діаграма декомпозиції роботи інтелектуального застосунку оцінки кредитоспроможності позичальника у нотації IDEF0

Сформуємо також діаграму у нотації EPC - event-driven process chain для демонстрації послідовності процесів та подій у межах веб-додатку (рисунок 2.12).

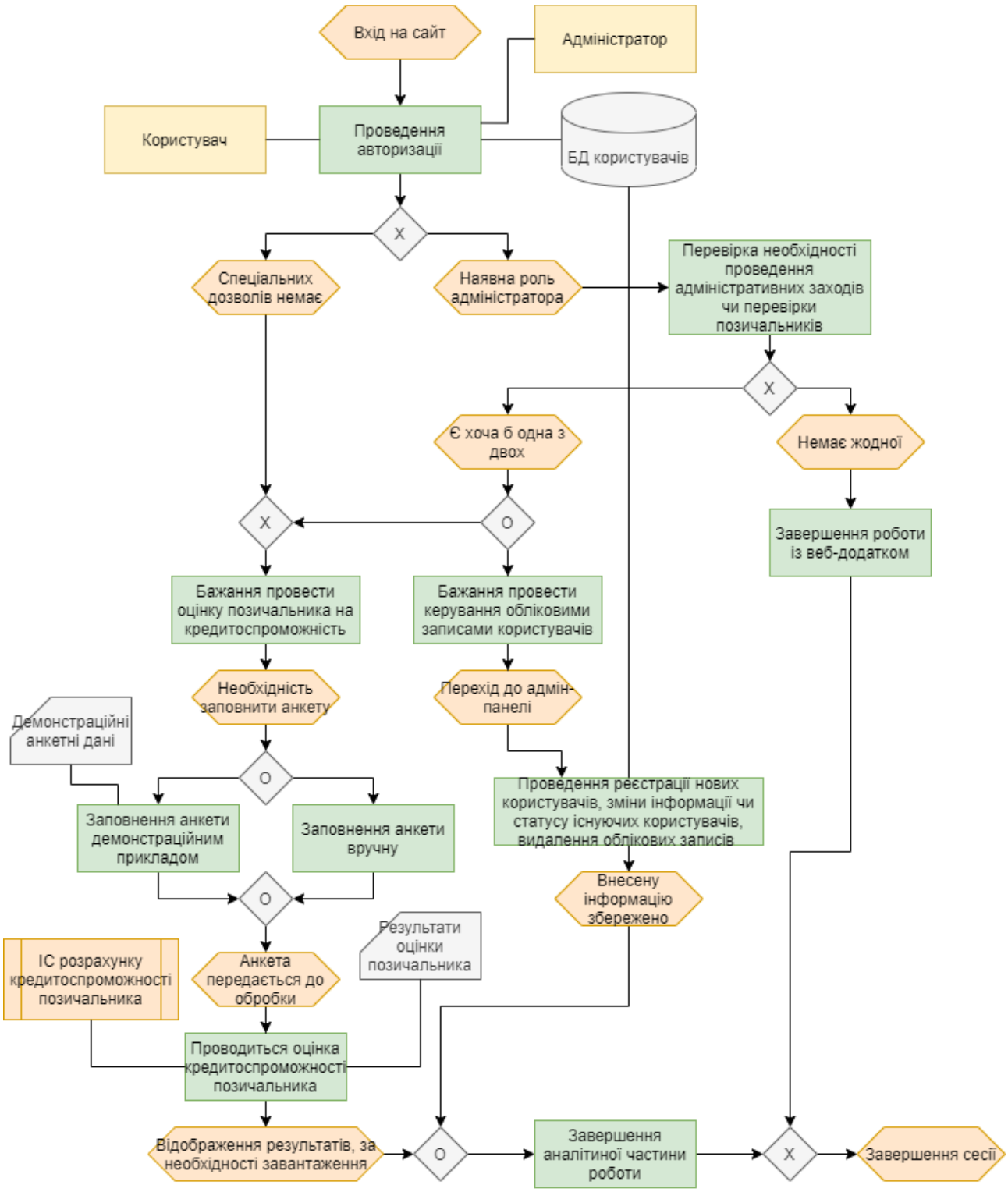


Рис 2.12 Процесна діаграма у нотації EPC.

Першим кроком є авторизація користувача. Неавторизовані користувачі не мають доступу до функцій застосунку. Далі, в залежності від рівня доступу. У випадку, якщо у користувача немає додаткових прав,

єдиною доступною функцією буде проведення оцінки кредитоспроможності. Для втілення цього потрібно заповнити форму про позичальника (або скористатися тестовим прикладом) та надіслати її на обробку. Далі йде етап відображення результатів та завершення роботи.

У випадку, коли користувач має права адміністратора, йому буде доступно перехід на адмін-панель для керування обліковими записами користувачів, або вихід з системи, якщо такої потреби поки немає.

2.4.3. Розробка інформаційного забезпечення

Нижче наведено логічну та фізичну моделі бази даних (рис 2.13а, 2.13б), спроектованої для даної роботи. Її необхідність обумовлюється тим, що має бути певна структура збереження та доступу до облікових записів користувачів веб-застосунку. Таким чином у цій реляційній базі даних, що називається «diploma», зберігається одне відношення. Цей опис додано для детальнішого розуміння праці системи, а також для можливості його подальшого розширення, наприклад, якщо збереження даних анкет перенести у цю базу даних. Таким чином поступово можливо буде накопичити нові дані для покращення моделі. Наразі подібне не реалізоване через малі обсяги системи і необхідність і так зберігати анкету із результатами для подальшої можливості завантаження. Зараз веб-застосунок ще не запущений на власному хості.

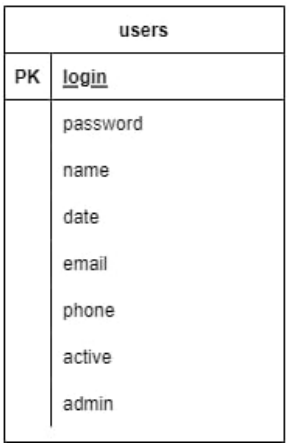


Рис 2.13а Даталогічна модель бази даних веб-застосунку

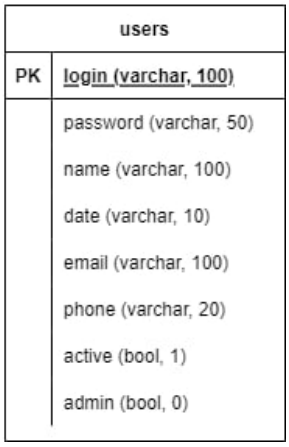


Рис 2.13б Фізична модель бази даних веб-застосунку

Окрім даних, що будуть доступні користувачеві до завантаження, а саме xlsx файл із анкетою та результами перевірки заявки та docx файл із звітом до дипломного проекту у ролі «Інструкції», проводиться збереження внутрішніх файлів у бінарному форматі. Прикладом подібних збережених файлів можуть слугувати файл із збереженою моделлю для того, щоб не навчати модель заново для оцінки кожної заявки, а також файл із параметрами попередньої обробки основного масиву даних. Якщо поглянути на структуру збереження даних, наведену нижче на рисунку 2.14, то можна помітити, що ці бінарні файли знаходяться у директорії python, далі директорії: encoders та models.

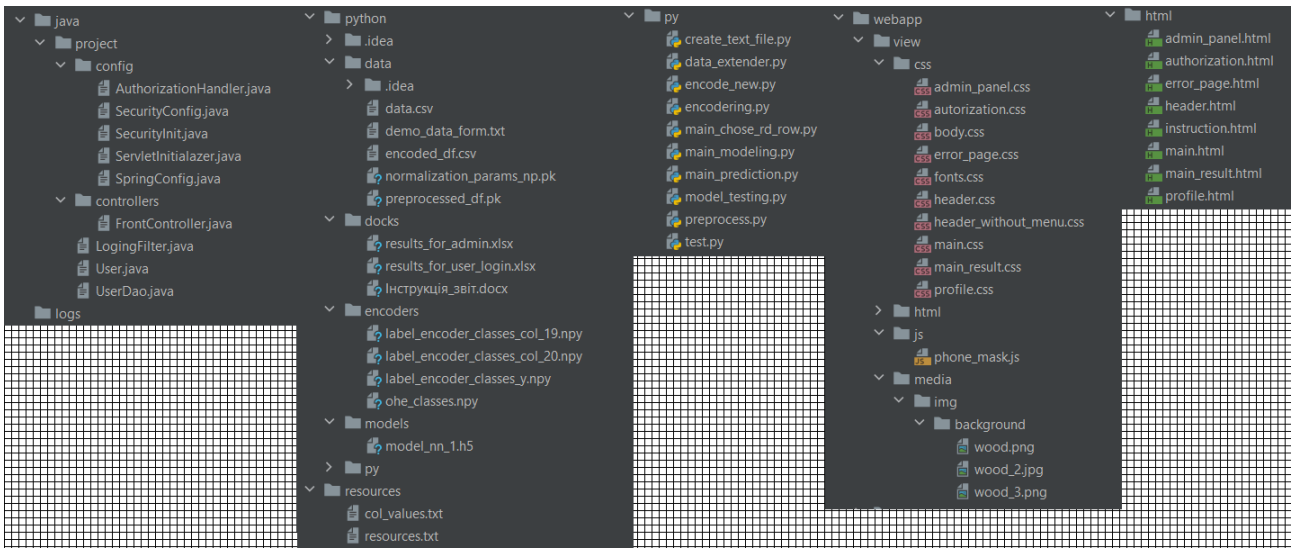


Рис 2.14 Структура збережених даних

На одному рівні у директорії проекту main знаходяться такі важливі директорії:

1) java - директорія для зберігання файлів керування серверною частиною веб-застосунку - логікою переходу за посиланнями, натискання та реакція кнопок, відправка запитів на виконання файлів python, взаємодія із базою даних користувачів, проведення авторизації, запис у змінні для відображення динамічного тексту на фронтенді.
2) python - директорія, в якій зберігається усе, із чим є взаємодія мовою програмування python, окрім папки resources. Тут знаходиться папка із даними для аналізу та побудови моделі, папка документів, в якій зберігається те, що користувачі можуть скачати за натисканням відповідної кнопки, папки із збереженими параметрами encoders та models.
3) py - директорія із основними розрахунковими файлами: енкодерінг даних, попередня обробка, моделювання, енкодерінг та обробка даних анкети, проведення самого аналізу анкети

4) resouces - директорія, із конфігурними файлами, щоб не шукати довго, де змінити одруковку.
5) webapp - директорія, що зберігає фронтенд. Тут знаходяться текстові сторінки веб-застосунку та стилі до них.

2.5. Попередня розробка інтерфейсу веб-додатку

За палітру взято такий набір кольорів та структур:

Рис 2.15 Палітра інтерфейсу веб-застосунку

Далі наведено попередньо розроблений основний макет інтерфейсу:

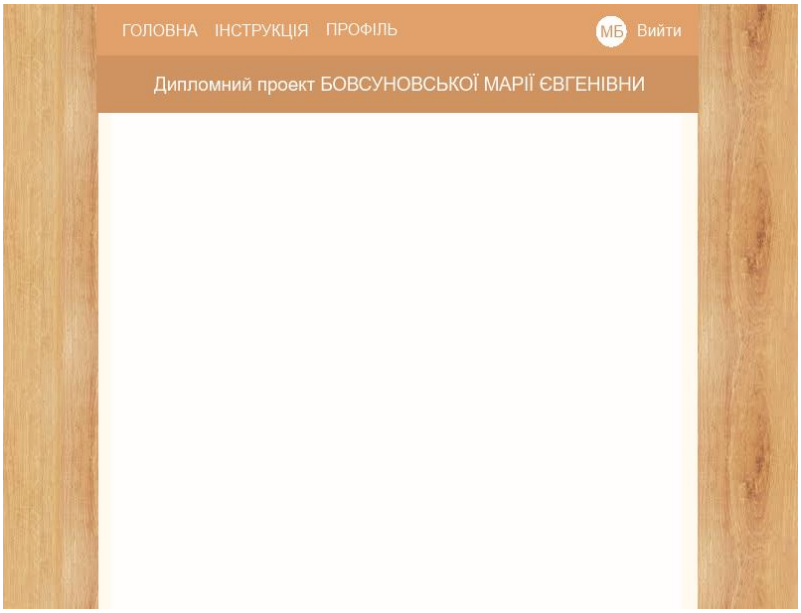


Рис 2.16 Основний макет ідеї дизайну веб-застосунку

Як можна помітити порівнявши рисунки 2.15 та 2.16, дерев’яна структура макету була замінена на ту, що наведена на рисунку 2.15. Зміни макету були обумовлені особливостями візуалізації при повторенні окремих сегментів зображення. Також варто звернути увагу, що шрифти тексту пізніше були замінені з тих, що наведені на макеті на Times New Roman з огляду на краще поєднання.

2.6. Висновки до розділу 2

Другий розділ даної дипломної роботи передбачає опис проектних рішень та теоретичної частини алгоритмів вирішення поставленої задачі класифікації.

Було детально оглянуто структуру набору даних, сформовано короткі теоретичні відомості про методи аналізу даних, які були використані у роботі. Було проведено планування та проектування архітектури системи оцінки кредитоспроможності позичальника, проведено детальний функціональний та процесний аналіз.

Розроблено та описано інформаційне забезпечення веб-застосунку.

Наведено етапи розробки інтерфейсу веб-застосунку.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЗНАЧЕННЯ КРЕДИТОСПРОМОЖНОСТІ ПОЗИЧАЛЬНИКА

3.1. Обрані засоби програмної розробки

Спираючись на архітектуру веб-додатку для оцінки кредитоспроможності клієнта на рисунку 2.8, було згруповано та розділено на групи наступні обрані засоби програмної розробки.

Для розробки програного модулю оцінки кредитоспроможності позичальників використано мову програмування Python - високорівневу інтерпретовану мову програмування, яка, порівняно із іншими, сильно пришвидшує написання програм за рахунок спрощення читабельності програмного коду. Найбільш важливими використаними бібліотеками Python можна згадати:

1) NumPy - пакет для математичних розрахунків;
2) Pandas - пакет для зручної роботи із багатовимірними масивами;
3) MatplotLib - пакет для візуалізації дво- та трьовимірних даних;
4) Sklearn - пакет для машинного навчання, включає дуже багато методів для підготовки та оцінки даних;
5) Tensorflow - пакет для машинного навчання та штучного інтелекту;
6) Keras - пакет для роботи із нейронними мережами.

Для формування інтерфейсу веб-додатку, без урахування динамічного вмісту сторінок, було застосовано такі засоби:

1) HTML5 - мова розмітки тексту для формування вмісту веб-сторінок;

2) CSS3 - мова задання стилю веб-сторінок;
3) JavaScript - мова програмування для створення інтерактивності веб-сторінок.

Для роботи із базою даних було застосовано:

1) DAO - патерн проектування, що використовується для спрощення доступу до інформації у базі даних;
2) MySQL - мова побудови запитів до бази даних.

У формуванні програмної частини сервісу було використано:

1) Java - об’єктно-орієнтована мова програмування, що тут використовувалася для роботи зі Spring;
2) Spring - фреймворк, написаний на мові Java, із якого використані такі модулі:
    1) Spring MVC - модуль Spring для створення веб-додатків, що використовує патерн додатків Model-View-Controller;
    2) Spring Security - модуль Spring для забезпечення автентифікації та контролю за доступом;
3) Thymeleaf - Java template engine, що використовується на стороні серверу для генерації змінного вмісту веб-сторінок (тут використовується саме для роботи із HTML5).

Важливі деталі програмного коду можна переглянути у ДОДАТКУ В.

3.2. Опис структури програмного продукту

Сформовано наступний перелік сторінок веб-додатку:

Сторінка авторизації (рівень доступу: Гість) - це сторінка, на яку потрапляють усі неавторизовані користувачі. Якщо сторінки за точним посиланням не існує, але вона знаходиться у межах домену, то користувач

теж потрапляє сюди.

Головна (рівень доступу: Користувач, Адміністратор) - сторінка із анкетою для визначення кредитоспроможності позичальника. На ній є можливість заповнити анкету вручну, внісши дані у поля по черзі, завантаження до полів демонстраційного прикладу, стирання усієї введеної інформації, якщо вона наприклад введена помилково. Окрім того на сторінці присутня кнопка проведення розрахунку - натиснути на неї можливо тільки за умови, якщо усі поля заповнені і заповнені коректно. Виконання запиту за кнопкою проведення розрахунку може зайняти певний час (не довше хвилини). Після виконання запиту - вміст сторінки зміниться на результати проведеного моделювання за внесеними даними. На сторінці буде відображено ідентифікаційні дані анкети та потенційного позичальника, а також імовірність того, що потенційний клієнт поверне даний кредит без урахування точності моделі. На цій версії сторінки кнопки управління замінюються на кнопки переходу до заповнення нової анкети та можливості завантаження анкети із результатами.

Профіль (рівень доступу: Користувач, Адміністратор) - сторінка, на якій відображаються реєстраційні дані наявного користувача, стрічка із паролем відсутня. Для користувачів із роллю адміністратора на цій сторінці також наявна кнопка переходу на сторінку «Адмін-панель».

Сторінка помилки (рівень доступу: Користувач, Адміністратор) - сторінка помилки, що генерує вміст в залежності від типу помилки. Для зручності присутня кнопка переходу на головну сторінку. Наразі передбачені для відображення помилки:

1) 404 - Ви знаходитеся тут тому, що сторінка за даною адресою не існує чи була переміщена на нове місце.
2) 403 - Право доступу до сторінки відхилено. Можливо вам треба змінити рівень доступу. Але скоріш за все, ви тут бути не

повинні.

3) 500 - Помилка пов'язана із внутрішньою проблемою сервера. Адміністрація скоріше за все вже знає, що сталося та намагається полагодити.

Адмін-панель (рівень доступу: Адміністратор) - сторінка на якій знаходиться таблиця-список усіх зареєстрованих користувачів. Є поля та кнопка під додавання нового користувача - вона знаходиться унизу списку. Адміністратор може у будь-якого користувача налаштувати статус доступу - надати рівень доступу адміністратора, видалити чи блокувати користувача. Заблоковані користувачі не мають доступу до сторінок веб-додатку.

Усі веб-сторінки, окрім сторінки помилки та сторінки авторизації, містять зверху шапку сайту. Для зручності шапку сайту за принципами ООР винесено у окремий файл. Використовуючи кнопки на у шапці сайту користувач може швидко та просто орієнтуватися сайтом. Окрім елементів переходу між сторінками та кнопки завантаження інструкції на шапці знаходиться кнопка виходу із облікового запису, при натисканні якої сесія користувача завершиться і його буде перенесено на сторінку авторизації. Кнопка «Інструкція» у шапці сайту виконує унікальну функцію - вона дозволяє завантажити звіт до цього дипломного проекту.

3.3. Попередня обробка даних та побудова моделі

Проведемо попередній аналіз даних. Зчитаємо дані та оглянемо отримане. На рисунку 3.1 показано зліва - типи даних та кількість об’єктів, справа - кількість унікальних елементів за атрибутом.

```
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
col_1     1000 non-null object
col_2     1000 non-null int64
col_3     1000 non-null object
col_4     1000 non-null object
col_5     1000 non-null int64
col_6     1000 non-null object
col_7     1000 non-null object
col_8     1000 non-null int64
col_9     1000 non-null object
col_10    1000 non-null object
col_11    1000 non-null int64
col_12    1000 non-null object
col_13    1000 non-null int64
col_14    1000 non-null object
col_15    1000 non-null object
col_16    1000 non-null int64
col_17    1000 non-null object
col_18    1000 non-null int64
col_19    1000 non-null object
col_20    1000 non-null object
y         1000 non-null int64
dtypes: int64(8), object(13)
```

```
col_1      4
col_2     33
col_3      5
col_4     10
col_5    921
col_6      5
col_7      5
col_8      4
col_9      2
col_10     3
col_11     4
col_12     4
col_13    53
col_14     3
col_15     3
col_16     4
col_17     4
col_18     2
col_19     2
col_20     2
y          2
```

Рис 3.1 Типи колонок даних, а також кількість унікальних значень по них

У ДОДАТКУ Б можна переглянути частоти та розподіли для кожного атрибута разом із типом даних та деякими статистичними даними по атрибуту. З рисунку 3.1 а також ДОДАТКУ Б можна помітити, що не усі зчитані дані відповідають зазначеному типу даних. Це пов’язано з тим, що більшість якісних (окрім 2, 5 та 13) факторів мають не більше чотирьох варіантів. Але це не є проблемою, адже ніхто не примушує їх розділяти на окремі бінарні стовпчики, як доведеться зробити із справді категоріальними факторами. Тим більше - зберігання числових даних у числовому вигляді збільшує стійкість системи. Нові введені дані, що пізніше будуть внесені до набору даних для переналаштування моделей машинного навчання краще б мали той самий тип.

Окрім того, у ДОДАТКУ Б важливо помітити, що цільова колонка «y» має незбалансовані класи, вплив цього фактору на проведення класифікації буде описано пізніше, але загалом це переводить задачу до майже зовсім іншого класу. Проведена заміна колонки «y» на бінарний показники - погано = 1, а добре = 0, для зручності. А також перекодовані буквені записи у цифри

шляхом додавання нових стовпчиків для кожного такого пункту із одиницею у відповідній комірці, а у всіх інших - з нулями. Для цього було застосовано one hot encoding. Для стовпчиків кількість унікальних значень у яких два було використано бінарне кодування за допомогою label encoding. Параметри для проведення декодування за тими ж правилами для анкети для передбачення збережено.

Пропущених значень, або рядків-дублів немає, тому подібні методи покращення та внесення змін до даних не мають необхідності бути використаними.

Для кількісних атрибутів необхідно провести нормалізацію. Як можна помітити із додатка 3.1 кількісні атрибути, що потребують нормалізації, а саме 2, 5 та 13 колонки, містять подібний до логарифмічного розподіл. Більшість значень зосереджена поблизу медіани, але є небагато великих значень, що не охоплюються діаграмою розмаху, ящиком із вусами, та вважаються за викиди. Очевидно, що коли третина значень - викиди, необхідно задуматися. Логарифмічна функція масштабування дозволяє уникнути втрат великих значень для подібних розподілів.

$$t_i = \frac{\log(x_i) - \log(\bar{x})}{\log(\max(x_i)) - \log(\min(x_i))}, \tag{3.1}$$

де t_i - нормалізоване значення

x_i - поки ще не нормалізоване значення

$\bar{x}$- середнє значення для розподілу

Таким чином проводиться нормалізація для стовпчиків 2,5 та 13. На рисунку 3.2. Перша колонка на рисунку показує розподіли без змін, друга - після нормалізації, третя - після прибирання викидів за методом IQR.

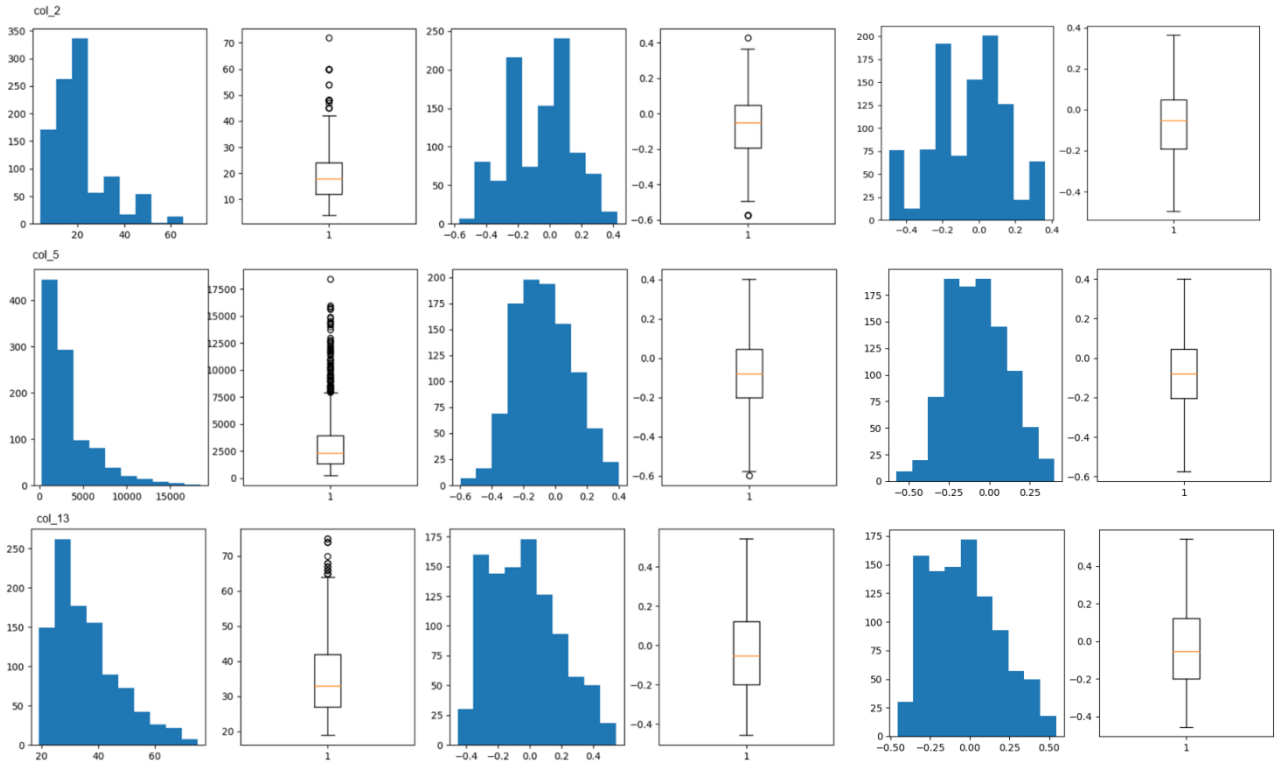


Рис 3.2 Проведення нормалізації для стовпчиків 2, 5 та 13. Прибирання викидів за методом IQR.

За атрибутом «Тривалість кредиту» було вилучено 7 викидів, за атрибутом «Сума кредиту» - 1, за атрибутом «Вік у роках» - не було визначено даних-викидів. Залишилося для аналізу 992 рядки.

Для розуміння того, як влаштовані дані, проведено кореляційний аналіз атрибутів, його зображено на рисунку 3.3.

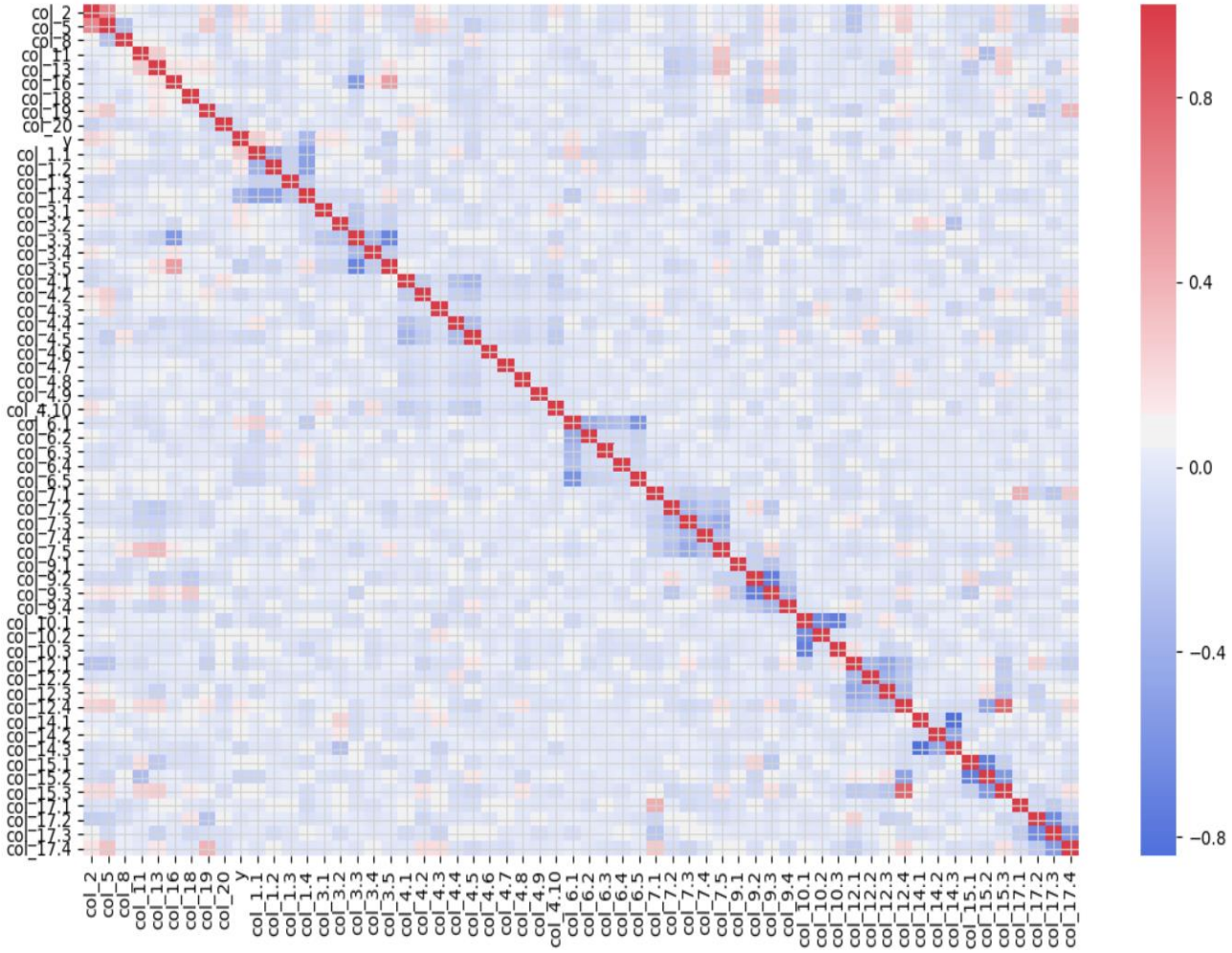


Рис 3.3 Кореляційна матриця атрибутів набору даних.

Серед кількісних атрибутів з рисунку 3.3 явно високу позитивну кореляцію мають між собою стовпчики 2 (Тривалість кредиту) та 5 (Сума кредиту). Значні негативні зв’язки поширені серед розпаралелених категоріальних атрибутів, адже, тому вони зазвичай і категоріальні, що коли людина підпадає під одну категорію - наприклад, має власний будинок), то вона вже не може входити у категорію людей, що орендують житло. Це логічно і нормально.

Параметрами для навчання штучної нейронної мережі стали параметри отримані в результаті великої кількості експериментальних прогонів навчання мережі. Але навіть із підібраними параметрами важливо розуміти, що набір даних для двадцяти атрибутів досить невеликий. Штучна нейронна

мережа має потенціал працювати із великими об’ємами даних дуже швидко, але коли будь-якій моделі недостатньо даних - необхідно зібрати більше і знову пройти усі ці етапи.

Для відбору найбільш впливових на цільовий стовпець атрибутів протестовано підходи використання різних метрик близькості, але сенс один - кореляція дуже слабка через невелику кількість даних. На даному етапі (із тими даними, які є) метриками оцінки кореляції до цільового стовпця можна оцінювати тільки те, наскільки багато нових даних додано до набору. Видалення навіть тих стовпців, що мають кореляцію із ціллю менше 0.05, впливає на точність класифікації і не впливає на час навчання моделі. Таким чином подальшим розвитком цього проекту могло би бути тестування на збільшених об’ємах даних. Важливо зауважити, що збільшення об’ємів тренувальних даних через методи ресемплінгу не показало жодного позитивного впливу.

Обрані дані через специфіку задачі, а саме критичну збитковість FN-помилок, а також небажаність TN-помилок, мають наступну матрицю вартостей: [28]

Таблиця 3.1 Матриця вартостей для задачі визначення кредитоспроможності позичальника [19]

	1(0)	2(1)
1(0)	0	2
2(1)	5	0

Параметри нейронної мережі в решті решт були обрані наступними:

1) Розмір batch: 50 - загальна кількість об’єктів для навчання, що обраховуються разом, чим більше це значення, тим більш загальним та згладженим буде результат, якщо це значення замаленьке, то результат хаотичний. Окрім того, за допомогою встановлення розміру батча є можливість не обробляти усі дані

разом.

2) Кількість епох навчання:15 - ітерації навчання нейронної мережі. Завелике число ітерацій може призвести до перенавчання, а замаленьке - до недонавчання.
3) Кількість прихованих шарів: 3 із власним параметром кількості нейронів - 50. Замаленька кількість нейронів чи шарів може призвести до недонавчання, а завелика - до перенавчання.
4) Ймовірність dropout : 0.3 - ймовірність того, що в даній ітерації випадковий нейрон не потрапить до тренування. Це методика, що зменшує перенавчання. [29]
5) Тип активаційної функції усіх прихованих шарів: LeakyRelu із власним параметром α=0.01. Активаційна функція - модифікація Relu, гарна для запобігання проблеми зникаючого градієнта. [30]
6) Активаційна функція вихідного шару: Sigmoid: активаційна функція, що використовується у випадках, коли задача для вирішення - задача бінарної класифікації.
7) Метод пошуку мінімуму втрат: Adam - Adaptive Moment Estimation, , метод адаптивної оцінки. У порівнянні із іншими методами показав себе значно краще на моїх даних. [31]

Початкова модель працює не дуже гарно із заданими параметрами, її показник f1 - наближений до 50%, навіть якщо accuracy наближена до 70%. Проблема у незбалансованості вибірки. Для незбалансованих даних - на точність ми орієнтуємося спочатку на метрику f1. Наступні методи ресемплінгу із найкращими підібраними експериментальним шляхом параметрами покращують критерій f1 приблизно на 7%:

RandomOverSampler, SMOTE, BorderlineSMOTE, SVMSMOTE, ADASYN.

Окрім того проведено тестування методу визначення ваг класів для

боротьби із незбалансованістю вибірки, кращих результатів із його використанням не отримано.

В решті решт застосовано метод ресемплінгу SVMSMOTE із параметрами k=10, m=10. Цей метод використовує ідею класифікатора SVM для генерації нових точок в околі. Він шукає опорні вектори і сторює синтетичні дані з їх урахуванням. [32]

Остаточна якість моделі перерахована за допомогою крос-валідації:

Таблиця 3.2 Параметри побудованої моделі, розраховані на тестових даних.

	mean	std
loss	60.208689	3.304802
accuracy	75.802752	3.241626
f1_score	56.924880	4.442272
precision	58.508437	3.656500
recall	58.034699	6.856125
AUC	76.977191	2.132649

3.4.Демонстрація прикладу роботи

Демонстрацію прикладу роботи наведено на послідовності дій з рисунку 2.12 - EPC діаграми.

Користувач потрапляє на сторінку веб-додатку у ролі Гостя і проводить авторизацію (рис 3.4).

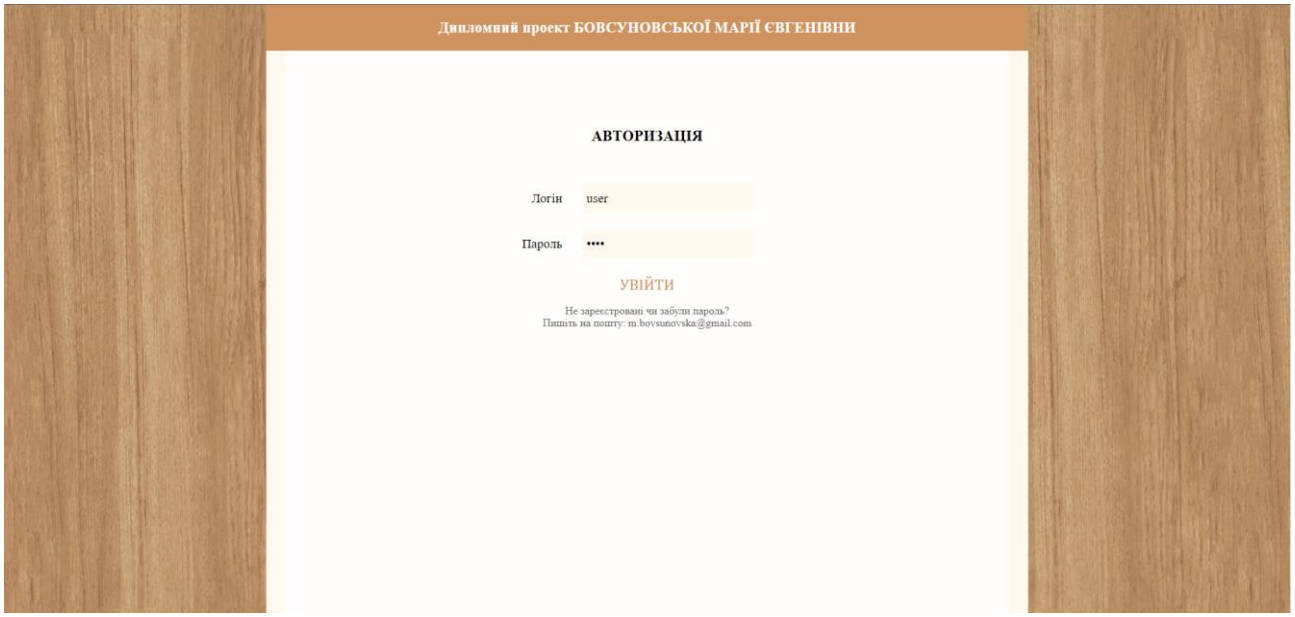


Рис 3.4 Сторінка авторизації

В залежності від ролі, яку його обліковий запис має, користувач перенаправляється на іншу сторінку. Якщо користувач - Звичайний користувач, то його перенаправляє на головну сторінку. Спочатку буде розглянута ситуація, коли користувач має роль - Адміністратор. Користувачів із такою роллю після авторизації перенаправляє на сторінку панелі адміністрування. Її наведено на рисунку 3.5

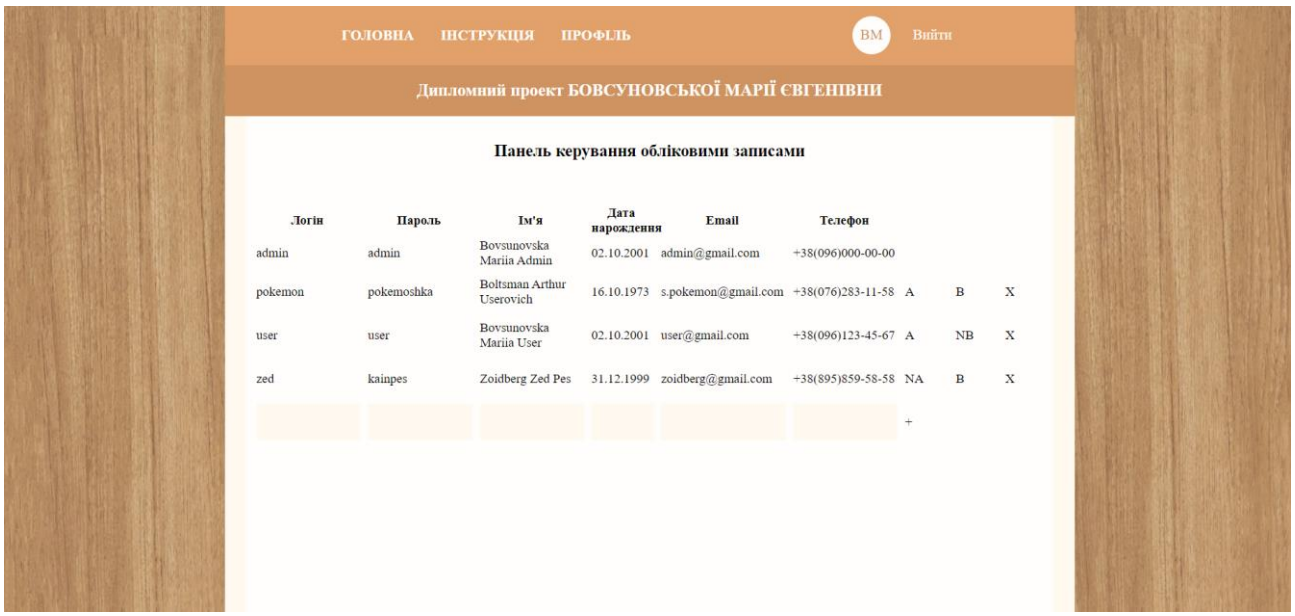


Рис 3.5 Сторінка панелі керування обліковими записами

Далі наведено інструкцію для керування обліковими записами у панелі

адміністратора.

Якщо у користувача із роллю адміністратора є необхідність керування обліковими записами, він може одразу тут додати нового користувача, заповнивши поля для даних знизу та після цього натиснувши на «+». Якщо поряд із стрічкою облікового запису є літера «A», то це означає, що користувач має звичайний обліковий запис, і, натиснувши на літеру, можна надати йому права адміністратора. Після цієї дії кнопка зміниться на «NA», що означає, що користувач отримав роль адміністратора і натиснувши на «NA» можна зробити його не адміністратором.

Аналогічно із кнопкою блокування доступу користувача - «B» означає зробити користувача заблокованим, «NB» - зробити користувача не заблокованим.

Кнопка «X» - протилежна кнопці «+» і відповідає за видалення користувача.

Видалити, заблокувати зробити не адміністраторським свій обліковий запис не є можливим. До адміністрування не допускаються не відповідальні особи.

Далі буде розглянута ситуація, коли користувач має роль - Адміністратора чи. Користувача. Із роллю Користувача користувача після авторизації перенаправляє одразу на головну сторінку (рис 3.6). Пізніше він, а також користувач із роллю адміністратора може перейти на неї у будь-який час.

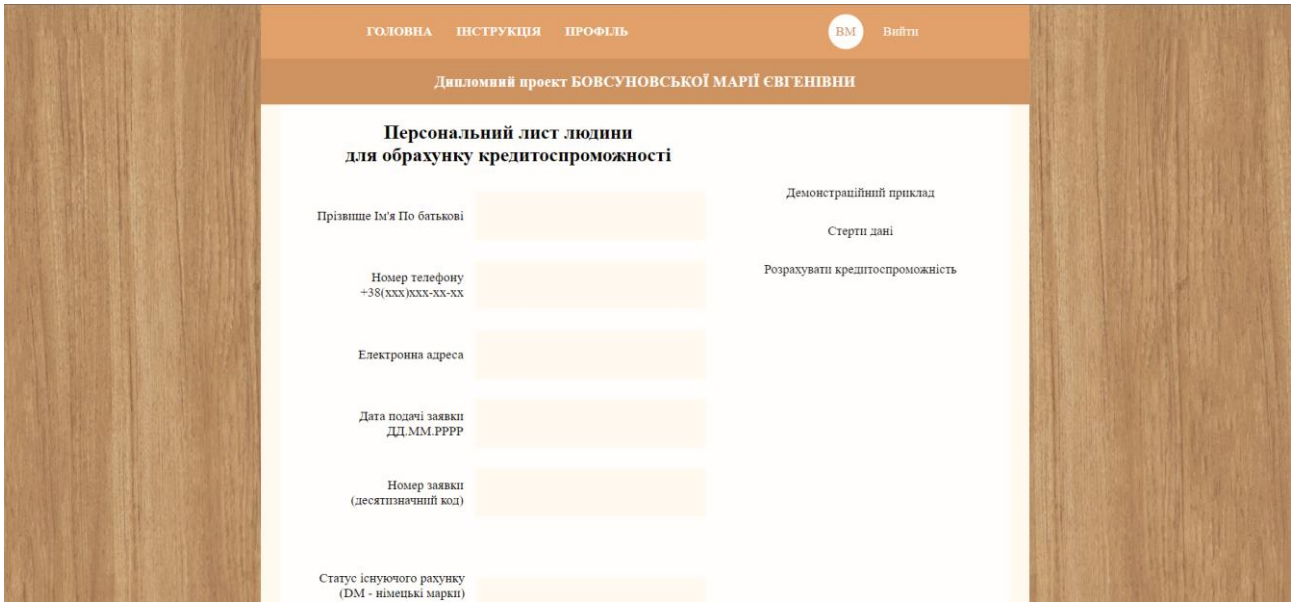


Рис 3.6 Головна сторінка.

Анкета на головній сторінці складається із 5 полів, що уніфікують її та 20 полів із характеристичними даними потенційного позичальника. Усі поля є обов’язковими. Для зручності вводу продумано введення необхідних типів даних, а для категоріальних атрибутів - випадаючі списки для вибору варіанта. (рис. 3.7)

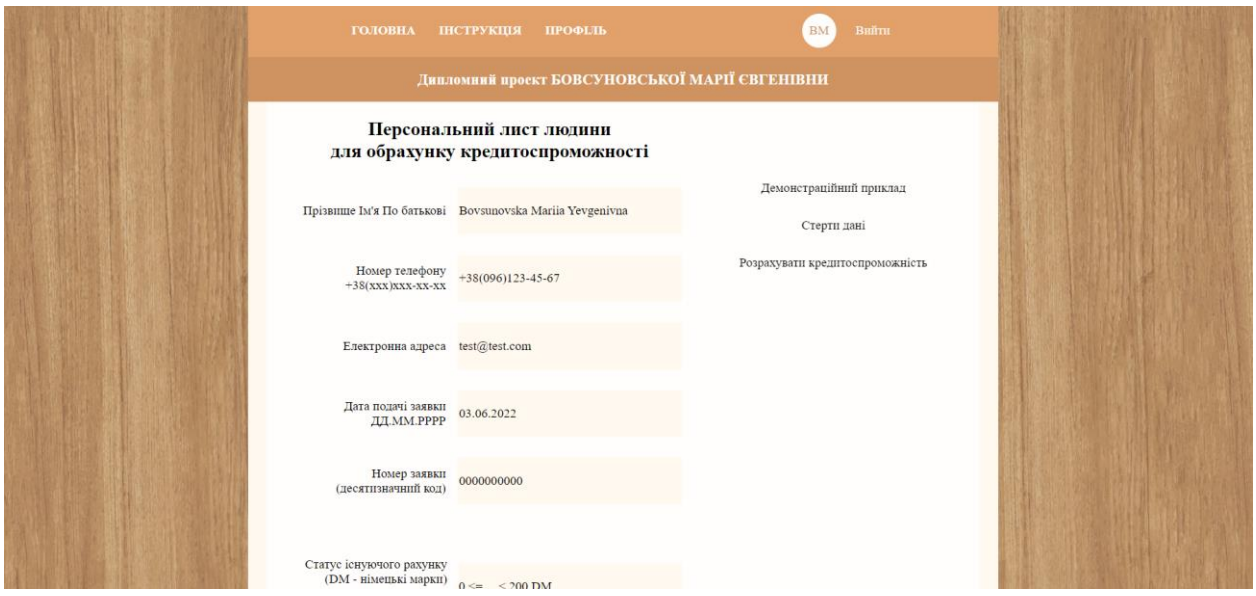


Рис 3.7 Головна сторінка. Заповнена демонстраційним прикладом.

Користувач також може стерти внесені дані натиснувши на кнопку Стерти дані. Натискання цієї кнопки, коли усі поля пусті, ні до чого не

приведе. Натискання кнопки розрахунку відправить дані із анкети до обрахунку. Цей процес може зайняти до хвилини. Після того, як сторінка із результатом завантажиться, на ній можна побачити введені раніше унікальні дані анкети, а також результати обробки даних, серед яких є відсоток того, що позичальник поверне кредит, без урахування точності моделі, і рекомендаційний опис ризикованості надання кредиту даному позичальнику. З цієї сторінки (рис 3.8) можна завантажити отримані результати у форматі для читання за допомогою Microsoft Excel.

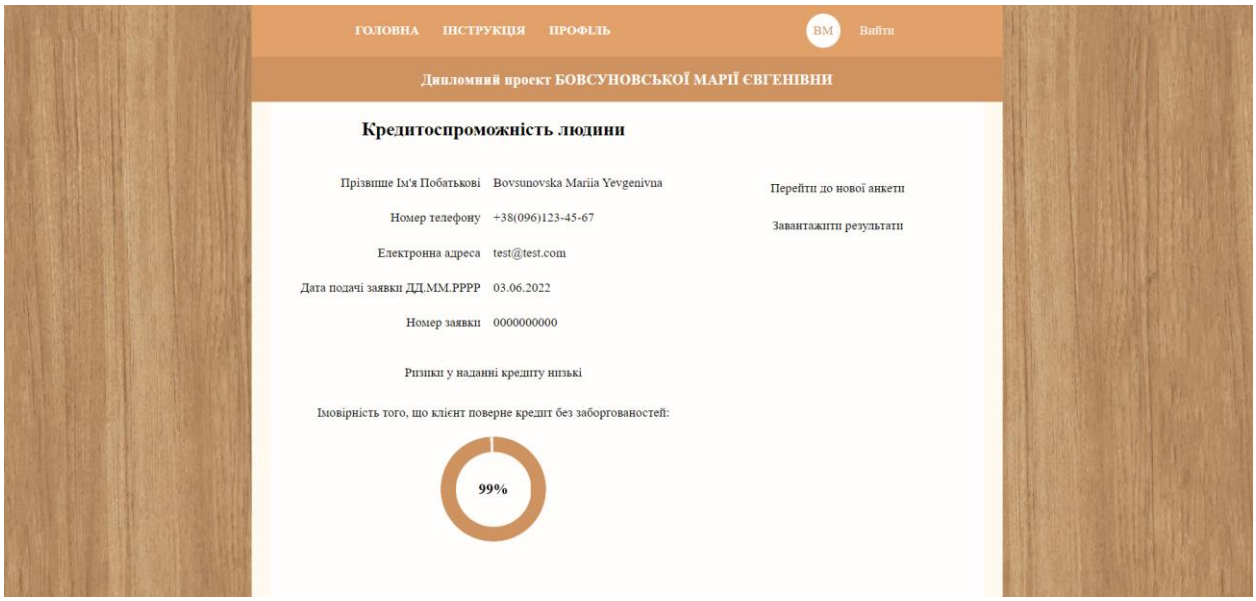
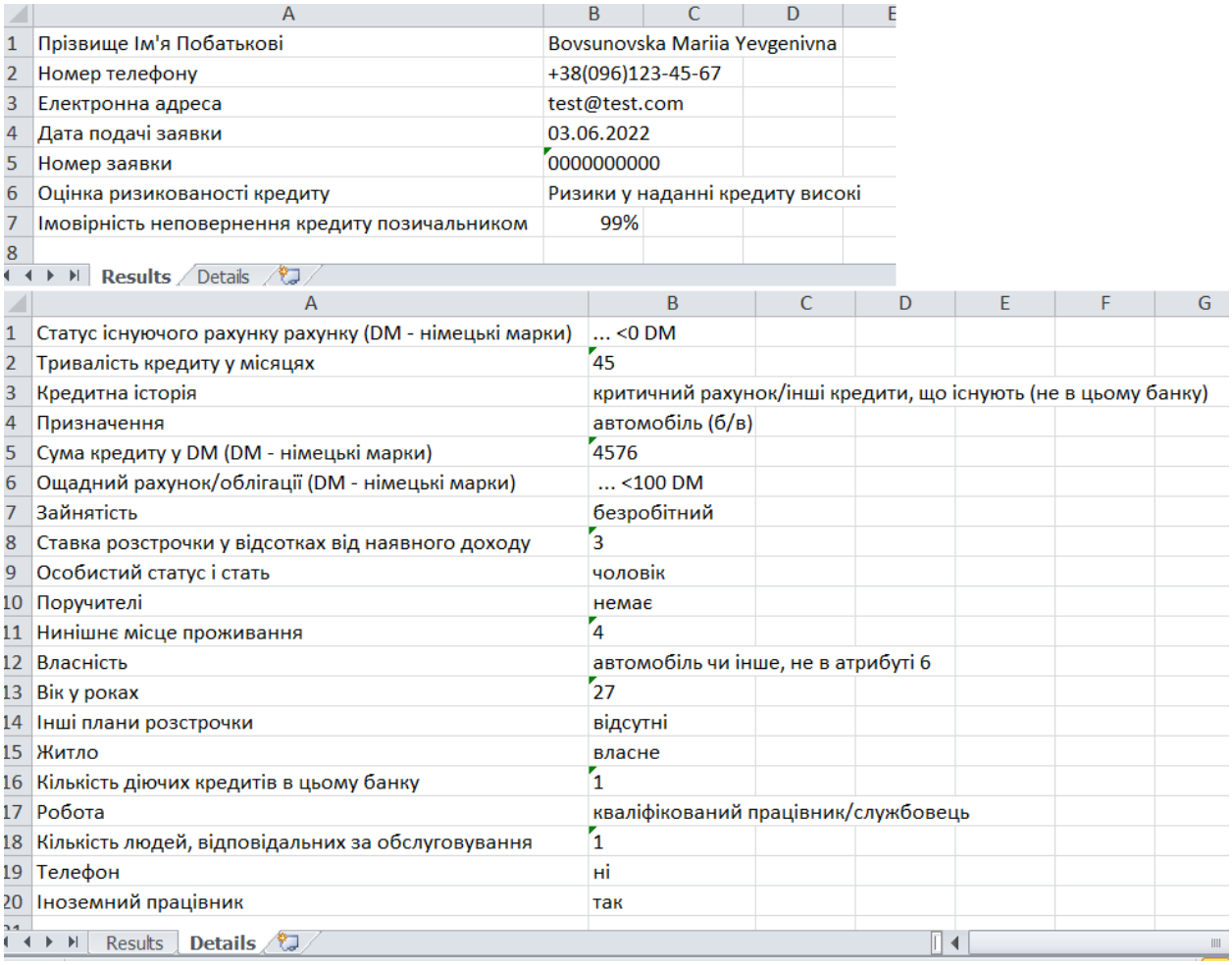


Рис 3.8 Головна сторінка. Результати проведеного аналізу анкети

Також можна завантажити результати проведеного аналізу у форматі xlsx. (рисунок 3.9), цей файл містить дві сторінки:



	A	B
1	Прізвище Ім'я Побаткові	Bovsunovska Mariia Yevgenivna
2	Номер телефону	+38(096)123-45-67
3	Електронна адреса	test@test.com
4	Дата подачі заявки	03.06.2022
5	Номер заявки	0000000000
6	Оцінка ризикованості кредиту	Ризики у наданні кредиту високі
7	Імовірність неповернення кредиту позичальником	99%
8		

Results | Details

	A	B
1	Статус існуючого рахунку рахунку (DM - німецькі марки)	... <0 DM
2	Тривалість кредиту у місяцях	45
3	Кредитна історія	критичний рахунок/інші кредити, що існують (не в цьому банку)
4	Призначення	автомобіль (б/в)
5	Сума кредиту у DM (DM - німецькі марки)	4576
6	Ощадний рахунок/облігації (DM - німецькі марки)	... <100 DM
7	Зайнятість	безробітний
8	Ставка розстрочки у відсотках від наявного доходу	3
9	Особистий статус і стать	чоловік
10	Поручителі	немає
11	Нинішнє місце проживання	4
12	Власність	автомобіль чи інше, не в атрибуті 6
13	Вік у роках	27
14	Інші плани розстрочки	відсутні
15	Житло	власне
16	Кількість діючих кредитів в цьому банку	1
17	Робота	кваліфікований працівник/службовець
18	Кількість людей, відповідальних за обслуговування	1
19	Телефон	ні
20	Іноземний працівник	так

Results | Details

Рис 3.9 Приклад розміщення інформації у збереженому xlsx файлі із результатами та анкетою.

Із будь-якого місця веб-додатку можна перейти до сторінки профілю. Звичайні користувачі тут можуть переглянути контактну інформацію, щоб вона не застарівала, а адміністратори зі сторінки профілю мають можливість перейти до адмін-панелі, яка була описана вище (рис 3.10).

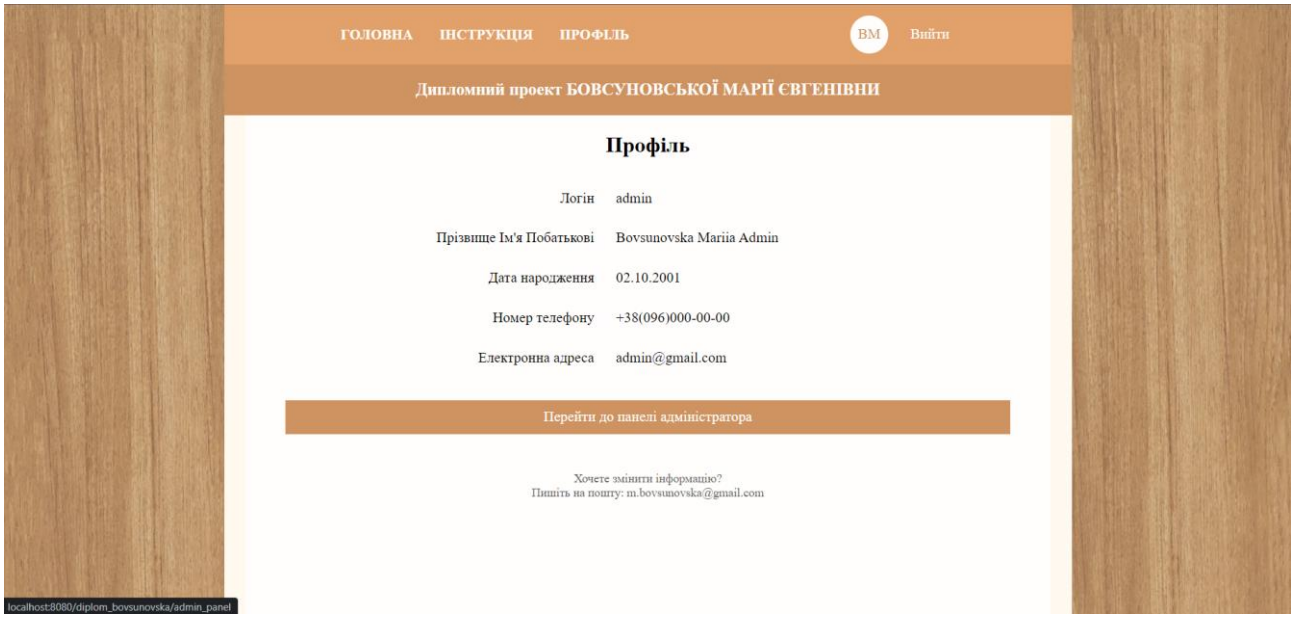


Рис 3.10 Сторінка профілю (кнопка переходу до адмін панелі доступна тільки адніністраторам)

Якщо спробувати наприклад надіслати запит на доступ до забороненого чи не існуючого місця, про відсутність доступу попіклується модуль Spring Sequrity, а про видачу помилки ось така наступна симпатична динамічна сторінка на рисунку 3.11.

Рис 3.11 Сторінка помилки. У даному випадку про відсутність сторінки за запитом, але текст та номер помилки можуть змінюватися.

3.5. Висновки до розділу 3

Третій розділ даної роботи передбачає розбір програмної реалізації інтелектуальної інформаційної системи вирішення сформульованої задачі проведення класифікації надання кредиту позичальнику.

У третьому розділі даної роботи було описано особливості програмної реалізації мети проекту - обґрунтовано вибір інструментів реалізації та параметрів моделей використаних алгоритмів, описано структуру програмного забезпечення та базу даних користувачів. Наведено приклади роботи веб-застосунку із описами можливих дій користувача.

ВИСНОВКИ

У ході даної роботи було розроблено інтелектуальну інформаційну систему, яка проводить скорингову оцінку платоспроможності позичальника за отриманими від нього даними та, після вирішення задачі класифікації, надає рекомендації про погодження чи відмову кредиту.

Були розглянуті варіанти, за яких причин гіпотетичний клієнт міг потрапити у проблемну ситуацію відмови у наданні позики, методики та проблеми оцінки кредитоспроможності позичальника, а також, яким чином на це впливає конкуренція між фінансово-кредитними установами. Були описані існуючі рішення та сформовано детальну постановку задачі для даної проблеми.

Було детально оглянуто структуру набору даних, сформовано короткі теоретичні відомості про методи аналізу даних, використані у роботі. Було проведено планування та проектування архітектури системи оцінки кредитоспроможності позичальника, проведено детальний функціональний та процесний аналіз, наведено етапи розробки інтерфейсу веб-додатку.

Було описано особливості програмної реалізації мети проекту - обґрунтовано вибір інструментів реалізації та параметрів моделей використаних алгоритмів, описано структуру програмного забезпечення та базу даних користувачів. Розроблено та описано інформаційне забезпечення веб-застосунку. Наведено етапи розробки інтерфейсу веб-застосунку.

Подальший розвиток тематики актуальний, адже поки існує позика, існують і ризики збитків, які мають бути оцінені та мінімізовані. Окрім того, що існує безліч алгоритмів машинного навчання, існує і безліч параметрів, які, в залежності від особливостей задачі, завжди будуть відрізнятися. Задачі швидкої та ефективної обробки великих масивів даних тільки набирають

популярності у сьогоденні, а зручний інтерфейс спрощує взаємодію із результатами. Саме тому розробка цієї дипломної роботи несе інтелектуальну цінність, адже тут детально розібрано предметну область та покроково описано та реалізовано інтелектуальну інформаційну систему вирішення задачі проведення класифікації надання кредиту позичальнику.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пахомова І.Г. Оцінка рівня конкурентності в банківській системі України / І. Г. Пахомова, Н. Т. Сірія // Ефективна економіка. – 2014. – No 6. – Режим доступу: http://www.economy.nayka.com.ua/?op=1&z=3106
2. Закон України «Про банки і банківську діяльність» [Електронний ресурс] // Відомості Верховної Ради України (ВВР). – 2001. – № 5-6, ст.30, із змінами внесеними згідно з Законами до 14.07.2021 включно. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2121-14/page
3. Огляд банківського сектору Лютий 2022 [Електронний ресурс] // Національний банк України. – 2022. – Режим доступу: https://bank.gov.ua/admin_uploads/article/Banking_Sector_Review_2022-02.pdf?v=4
4. Нагляд за ринком небанківських фінансових послуг [Електронний ресурс] // Національний банк України. – Режим доступу: https://bank.gov.ua/ua/supervision/split/registers-lists (дата звернення 07.02.22)
5. Сало І.В. Система управління конкурентоспроможністю банку / І.В. Сало, О.В. Мірошниченко // Актуальні проблеми економіки. – 2012. – № 5. – С. 279-286. Режим доступу: http://nbuv.gov.ua/UJRN/ape_2012_5_36
6. Руда О.Л. Сучасний стан банківської системи України та її конкурентоспроможність [Електронний ресурс] / Руда О.Л. // Ефективна економіка. – №4. – 2019. – Режим доступу: http://www.economy.nayka.com.ua/pdf/4_2019/63.pdf
7. Кльоба Р.Л. Маркетинговий підхід до вдосконалення управління банківською діяльністю / Кльоба Р.Л. // Наук. вісник НЛТУ України. – Вип. 19.3. – 2012. – С. 196–214 Режим доступу: https://cyberleninka.ru/article/n/marketingoviy-pidhid-do-vdoskonalennya-upravlinnya-bankivskoyu-diyalnistyu
8. Закон України «Про споживче кредитування» [Електронний ресурс] // Відомості Верховної Ради України (ВВР). – 2017. – № 1, ст.2, із змінами внесеними згідно з Законами до 14.07.2021 включно. – Режим доступу: https://zakon.rada.gov.ua/laws/show/1734-19
9. Bank nonperforming loans to total gross loans (%) [Електронний ресурс] // Worldbank. –

2020. – Режим доступу: https://data.worldbank.org/indicator/FB.AST.NPER.ZS?end=2020&start=2020&view=map&year=2020 (дата звернення 14.02.22)

10. Частка непрацюючих кредитів (NPL) висока, але поступово скорочується [Електронний ресурс] // Національний банк України. – 2021. – Режим доступу: https://bank.gov.ua/ua/stability/npl

11. Єдиний реєстр бюро кредитних історій [Електронний ресурс] // Режим доступу: https://data.gov.ua/dataset/bki-nfp (дата звернення 17.02.22)

12. Закон України «Про організацію формування та обігу кредитних історій» [Електронний ресурс] // Відомості Верховної Ради України (ВВР). – 2005. – № 32, ст.421, із змінами внесеними згідно із Законами до 15.09.2020 включно. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2704-15

13. Закредитованість населення України: 2016–2017 [Електронний ресурс] // Національний банк України: Аналітичний звіт. – 2017. – Режим доступу: https://nabu.ua/images/tinymce/file/IFC_GfK_Over_ind_ukr%20%281%29.pdf

14. Н.П. Шульга Інтегрована система управління ризиками банку: монографія [Електронний ресурс] / Н.П. Шульга, В.І. Міщенко, Л.Л. Анісімова та ін. // Київ: Київ. нац. торг.-екон. ун-т. – 2018. – 440 с. Режим доступу: https://knute.edu.ua/file/NjY4NQ==/293ad3f2051461a26841bc92ea06a800.pdf

15. Публікації [Електронний ресурс] // Національний банк України. – Режим доступу: https://bank.gov.ua/ua/publications (дата звернення 18.02.22)

16. О.В.Дзюблюк Кредитний ризик і ефективність діяльності банку: монографія / О.В.Дзюблюк, Л.М.Прийдун. // Тернопіль: ФОП Паляниця В.А. – 2015. – 295 с. – Режим доступу: https://bit.ly/3gYzF3p

17. Заявочний скоринг [Електронний ресурс] // Українське бюро кредитних історій. – Режим доступу: https://www.ubki.ua/appscore-ua (дата звернення 19.02.2022)

18. M.R.Kumar Review of Machine Learning models for Credit Scoring Analysis / M.R.Kumar, V.K.Gunjan // Revista Ingeniería Solidaria. – vol. 16. – no. 1. – 2020. – Режим доступу: https://doi.org/10.16925/2357-6014.2020.01.11

19. Statlog (German Credit Data) Data Set [Електронний ресурс] // UCI Machine Learning Repository. – 2000. – Режим доступу: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

20. Шлезінгер М. Десять лекцій по статистичному і структурному розпізнаванню / Шлезінгер М., Главач В. // Київ: Наукова думка. – 2004. – Режим доступу: http://www.irtc.org.ua/image/Files/Schles/esh10_full.pdf

21. D.Pyle. Data Preparation for Data Mining // Los Altos, California: Morgan Kaufmann Publishers. – 1999. – Режим доступу: https://archive.org/details/datapreparationf0000pyle

22. Верес О. М. Класифікація методів аналізу Великих даних / О.М. Верес, Р.М. Оливко // Вісник Національного університету «Львівська політехніка» . – Серія : Інформаційні системи та мережі. – 2017. – № 872. – С. 84-92. – Режим доступу: http://nbuv.gov.ua/UJRN/VNULPICM_2017_872_12

23. D.Çavuşoğlu Backpropagation paper from scratch [Електронний ресурс] / D.Çavuşoğlu // Towards Data Science. – 2020. – Режим доступу: https://towardsdatascience.com/backpropagation-paper-from-scratch-796793789248

24. M.H.Sazli A brief review of feed-forward neural networks [Електронний ресурс] / M.H.Sazli // Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering. – 2006. – №50 (01). – pp.11-17. – Режим доступу: https://dergipark.org.tr/en/download/article-file/1615327

25. J.Brownlee How to Choose a Feature Selection Method For Machine Learning [Електронний ресурс] / J.Brownlee // 2020. – Режим доступу: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

26. Селезньов Н.П Кореляційний аналіз навчального процесу на прикладі підсумкових оцінок учнів [Електронний ресурс] / Селезньов Н.П., Селезньова Н.В., Селезньов С.В. // Вісник НТУУ “КПІ”. Філософія. Психологія. Педагогіка. – №1. – 2012. – С. 139-145. – Режим доступу: https://ela.kpi.ua/bitstream/123456789/8468/1/30.pdf

27. Compare over-sampling samplers [Електронний ресурс] // Довідкова документація. Бібліотека Python Imblearn. – Режим доступу: https://imbalanced-learn.org/stable/auto_examples/over-sampling/plot_comparison_over_sampling.html

28. J.Brownlee Cost-Sensitive Learning for Imbalanced Classification [Електронний ресурс] / J.Brownlee // 2020. – Режим доступу: https://machinelearningmastery.com/cost-sensitive-learning-for-imbalanced-classification/

29. Machine learning with Tensorflow – Dropout [Електронний ресурс] // 2021. – Режим доступу: https://naolin.medium.com/machine-learning-with-tensorflow-dropout-ef15213f642f

30. R.Pramoditha How to Choose the Right Activation Function for Neural Networks [Електронний ресурс] / R.Pramoditha // Towards Data Science. – 2019. – Режим доступу: https://towardsdatascience.com/how-to-choose-the-right-activation-function-for-neural-networks-3941ff0e6f9c

31. S.Doshi Various Optimization Algorithms For Training Neural Network [Електронний ресурс] / S.Doshi // Towards Data Science. – 2019. – Режим доступу: https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6

32. Oversamplings methods. SVMSMOTE [Електронний ресурс] // Довідкова документація. Бібліотека Python Imblearn. – Режим доступу: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SVMSMOTE.html

ДОДАТКИ

ДОДАТОК А

Атрибути набору даних

атрибут_1: (я)	атрибут_2: (к)	атрибут_3: (я)	атрибут_4: (я)
Статус існуючого рахунку (DM - німецькі марки)	Тривалість кредиту у місяцях	Кредитна історія	Призначення
A11: ... <0 DM A12: 0 <= ... <200 DM A13: ...> = 200 DM/посадові оклади не менше 1 року A14: рахунок не перевірявся		A30: кредити не взяті/всі кредити повернені належним чином A31: всі кредити в цьому банку повернені належним чином A32: існуючі кредити повернені належним чином A33: затримка з виплатою в минулому A34: критичний рахунок/інші кредити, що існують (не в цьому банку)	A40: автомобіль (новий) A41: автомобіль (б/в) A42: меблі/обладнання A43: радіо/телебачення A44: побутова техніка A45: ремонт A46: освіта A48: перепідготовка A49: бізнес A410: відпустка/інші

Продовження ДОДАТКУ А

атрибут_5: (к)	атрибут_6: (я)	атрибут_7: (я)	атрибут_8: (к)
Сума кредиту у DM (DM - німецькі марки)	Ощадний рахунок/облігації (DM - німецькі марки)	Зайнятість	Ставка розстрочки у відсотках від наявного доходу
	A61: ... <100 DM A62: 100 <= ... <500 DM A63: 500 <= ... <1000 DM A64: ...> = 1000 DM A65: невідомо/немає ощадного рахунку	A71: безробітний A72: ... <1 рік A73: 1 <= ... <4 років A74: 4 <= ... <7 років A75 : ..> = 7 років	

атрибут_9: (я)	атрибут_10: (я)	атрибут_11: (к)	атрибут_12: (я)
Стать	Поручителі	Нинішнє місце проживання	Власність
A91: чоловік A92: жінка	A101: немає A102: спів-заявник A103: гарант		A121: нерухомість A122: (якщо не A121) угода про заощадження у житлово-будівельному кооперативі/страхування життя A123: (якщо не A121/A122) автомобіль чи інше, не на ощадному рахунку A124: невідомо/немає власності

Продовження ДОДАТКУ А

атрибут_13: (к)	атрибут_14: (я)	атрибут_15: (я)	атрибут_16: (к)
Вік у роках	Інші плани розстрочки	Житло	Кількість діючих кредитів в цьому банку
	A141: банк A142: магазини A143: відсутні	A151: оренда A152: власне A153: безкоштовно	

атрибут_17: (я)	атрибут_18: (к)	атрибут_19: (я)	атрибут_20: (я)
Робота	Кількість людей, відповідальних за обслуговування	Телефон	Іноземний працівник
A171: безробітний /некваліфікований (нерезидент) A172: некваліфікований (резидент) A173: кваліфікований працівник/службовець A174: менеджмент /самозайнята особа /висококваліфікований працівник/ службова особа		A191: немає A192: є, зареєстровано під іменем клієнта	A201: так A202: ні

ДОДАТОК Б

Частоти та розподіли для кожного атрибута набору даних

col_1
Categorical

Distinct count	4
Unique (%)	0.4%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A14	394	39.4%
A11	274	27.4%
A12	269	26.9%
A13	63	6.3%

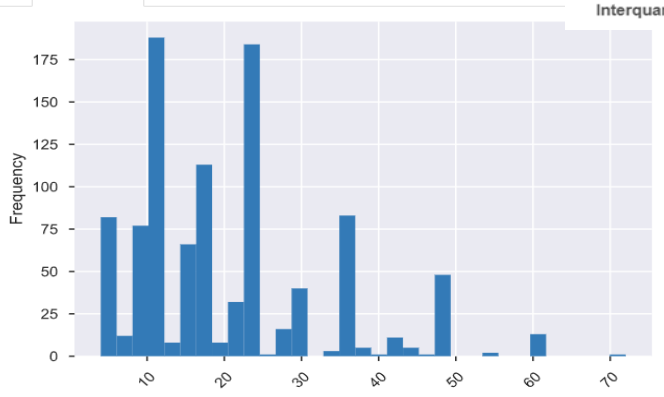
col_2
Numeric

Distinct count	33	Mean	20.903
Unique (%)	3.3%	Minimum	4
Missing (%)	0.0%	Maximum	72
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Quantile statistics

Minimum	4
5-th percentile	6
Q1	12
Median	18
Q3	24
95-th percentile	48
Maximum	72
Range	68
Interquartile range	12

Statistics | Histogram | Common values | Extreme values



col_3
Categorical

Distinct count	5
Unique (%)	0.5%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A32	530	53.0%
A34	293	29.3%
A33	88	8.8%
A31	49	4.9%
A30	40	4.0%

Продовження ДОДАТКУ Б

col_4
Categorical

Distinct count	10
Unique (%)	1.0%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

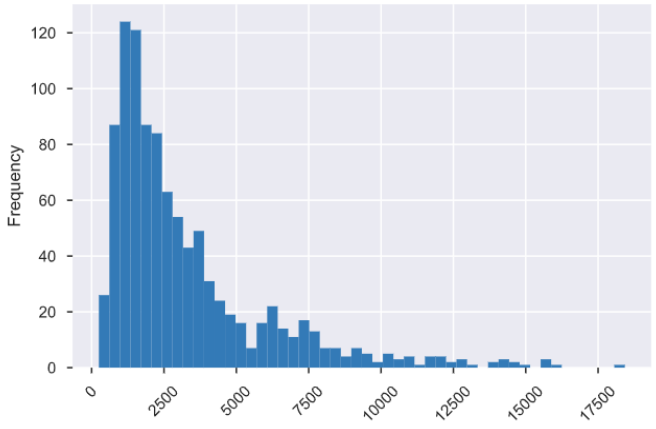
Value	Count	Frequency (%)
A43	280	28.0%
A40	234	23.4%
A42	181	18.1%
A41	103	10.3%
A49	97	9.7%
A46	50	5.0%
A45	22	2.2%
A44	12	1.2%
A410	12	1.2%
A48	9	0.9%

col_5
Numeric

Distinct count	921	**Mean**	3271.258
Unique (%)	92.1%	**Minimum**	250
Missing (%)	0.0%	**Maximum**	18424
Missing (n)	0	**Zeros (%)**	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Toggle details

Statistics | Histogram | Common values | Extreme values



Quantile statistics

Minimum	250
5-th percentile	708.95
Q1	1365.5
Median	2319.5
Q3	3972.25
95-th percentile	9162.7
Maximum	18424
Range	18174
Interquartile range	2606.75

Продовження ДОДАТКУ Б

col_6
Categorical

Distinct count	5
Unique (%)	0.5%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A61	603	60.3%
A65	183	18.3%
A62	103	10.3%
A63	63	6.3%
A64	48	4.8%

col_7
Categorical

Distinct count	5
Unique (%)	0.5%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A73	339	33.9%
A75	253	25.3%
A74	174	17.4%
A72	172	17.2%
A71	62	6.2%

col_8
Categorical

Distinct count	4
Unique (%)	0.4%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
4	476	47.6%
2	231	23.1%
3	157	15.7%
1	136	13.6%

Продовження ДОДАТКУ Б

col_9
Categorical

Distinct count	2
Unique (%)	0.2%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A91	690	69.0%
A92	310	31.0%

col_10
Categorical

Distinct count	3
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A101	907	90.7%
A103	52	5.2%
A102	41	4.1%

col_11
Categorical

Distinct count	4
Unique (%)	0.4%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
4	413	41.3%
2	308	30.8%
3	149	14.9%
1	130	13.0%

col_12
Categorical

Distinct count	4
Unique (%)	0.4%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A123	332	33.2%
A121	282	28.2%
A122	232	23.2%
A124	154	15.4%

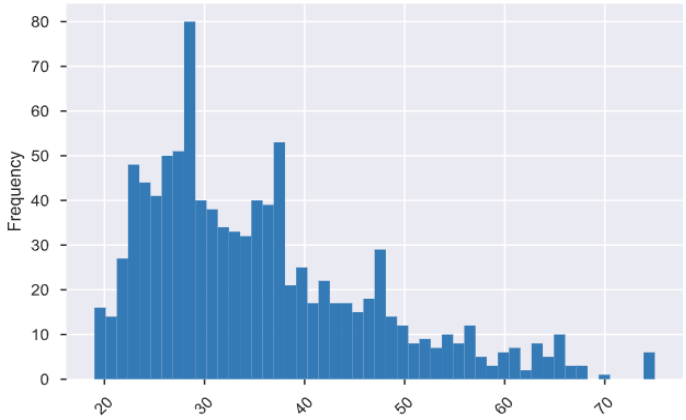
Продовження ДОДАТКУ Б

col_13
Numeric

Distinct count	53	**Mean**	35.546	**Minimum**	19
Unique (%)	5.3%	**Minimum**	19	**5-th percentile**	22
Missing (%)	0.0%	**Maximum**	75	**Q1**	27
Missing (n)	0	**Zeros (%)**	0.0%	**Median**	33
Infinite (%)	0.0%			**Q3**	42
Infinite (n)	0			**95-th percentile**	60
				Maximum	75
				Range	56
				Interquartile range	15

Quantile statistics

Statistics | Histogram | Common values | Extreme values



col_14
Categorical

Distinct count	3
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A143	814	81.4%
A141	139	13.9%
A142	47	4.7%

col_15
Categorical

Distinct count	3
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A152	713	71.3%
A151	179	17.9%
A153	108	10.8%

Продовження ДОДАТКУ Б

col_16
Categorical

Distinct count	4
Unique (%)	0.4%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
1	633	63.3%
2	333	33.3%
3	28	2.8%
4	6	0.6%

col_17
Categorical

Distinct count	4
Unique (%)	0.4%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A173	630	63.0%
A172	200	20.0%
A174	148	14.8%
A171	22	2.2%

col_18
Categorical

Distinct count	2
Unique (%)	0.2%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
1	845	84.5%
2	155	15.5%

Продовження ДОДАТКУ Б

col_19
Categorical

Distinct count	2
Unique (%)	0.2%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A191	596	59.6%
A192	404	40.4%

col_20
Categorical

Distinct count	2
Unique (%)	0.2%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
A201	963	96.3%
A202	37	3.7%

y
Categorical

Distinct count	2
Unique (%)	0.2%
Missing (%)	0.0%
Missing (n)	0

Toggle details

Common values | Composition

Value	Count	Frequency (%)
1	700	70.0%
2	300	30.0%

ДОДАТОК В

Лістинг програмного коду

encodering.py - відповідає за проведення енкодерінгу вхідних даних

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from pickle import dump as pdump


def encode_main_data(data_adress):
    # DATA reading
    df = pd.read_csv(data_adress, sep=";", header=0, index_col=False)

    # df.info()
    # print("\n", df[:5])

    # encodering
    toLabel = ["col_19", "col_20", "y"]
    toOne = ["col_1", "col_3", "col_4", "col_6", "col_7", "col_9", "col_10", "col_12", "col_14", "col_15",
"col_17"]
    newNames = ["col_2", "col_5", "col_8", "col_11", "col_13", "col_16", "col_18", "col_19", "col_20", "y"
        , "col_1.1", "col_1.2", "col_1.3", "col_1.4"
        , "col_3.1", "col_3.2", "col_3.3", "col_3.4", "col_3.5"
        , "col_4.1", "col_4.2", "col_4.3", "col_4.4", "col_4.5", "col_4.6", "col_4.7", "col_4.8", "col_4.9",
"col_4.10"
        , "col_6.1", "col_6.2", "col_6.3", "col_6.4", "col_6.5"
        , "col_7.1", "col_7.2", "col_7.3", "col_7.4", "col_7.5"
        , "col_9.1", "col_9.2"
        , "col_10.1", "col_10.2", "col_10.3"
        , "col_12.1", "col_12.2", "col_12.3", "col_12.4"
        , "col_14.1", "col_14.2", "col_14.3"
        , "col_15.1", "col_15.2", "col_15.3"
        , "col_17.1", "col_17.2", "col_17.3", "col_17.4"]

    # ініціалізуємо енкодери
    encoder_l = LabelEncoder()
    encoder_ohe = OneHotEncoder(sparse=False)

    encoded_categorical_columns = pd.DataFrame(encoder_ohe.fit_transform(df[toOne]))
    with open('../encoders/ohe_classes.npy', 'wb') as fi:
        pdump(encoder_ohe, fi)

    for i in toLabel:
        df[i] = encoder_l.fit_transform(df[i])
        with open('../encoders/label_encoder_classes_'+i+'.npy', 'wb') as fi:
            pdump(encoder_l, fi)

    df = df.drop(toOne, axis=1)
    df = pd.DataFrame(np.hstack((df, encoded_categorical_columns)))
    df.columns = [newNames]

    # пересуваємо y в кінець для зручності
    tmp_y = df['y'].copy()
```

Продовження ДОДАТКУ В

```
    df.pop('y')
    df['y'] = tmp_y

    # to_file
    df.to_csv("../data/encoded_df.csv", index=None, header=True)
```

preprocess.py - відповідає за проведення нормалізації. видалення викидів

```
import pandas as pd
import numpy as np
from pickle import dump as pdump
import matplotlib.pyplot as plt
from encodering import encode_main_data


def preprocess():
    # проводимо енкодінг даних
    encode_main_data("../data/data.csv")

    # завантажуємо декодовані дані
    en_df = pd.read_csv("../data/encoded_df.csv", sep=",", header=0, index_col=False)
    pd.options.display.max_columns = 21
    toSt = ["col_2", "col_5", "col_13"]

    en_dflog = en_df.copy()
    normalization_params=[]
    for i in toSt:
        normalization_params.append([np.mean(en_df[i]),max(en_df[i]),min(en_df[i])])
        en_dflog[i] = (np.log2(en_df[i]) - np.log2(normalization_params[toSt.index(i)][0])) / \
                (np.log2(normalization_params[toSt.index(i)][1]) -
np.log2(normalization_params[toSt.index(i)][2]))
        # print(i)
        # fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize=(15, 4))
        # ax1.hist(en_df[i])
        # ax2.boxplot(en_df[i])
        # ax3.hist(en_dflog[i])
        # ax4.boxplot(en_dflog[i])
        # plt.show()

        en_df[i] = en_dflog[i]

    with open('../data/normalization_params_np.pk', 'wb') as fi:
        # dump your data into the file
        pdump(normalization_params, fi)

    # vydalennya vybrosiv iqr
    for i in toSt:
        # print(i)
        # fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize=(15, 4))
        # ax1.hist(en_df[i])
        # ax2.boxplot(en_df[i])

        q25, q75 = np.percentile(en_df[i], 25), np.percentile(en_df[i], 75)
        iqr = q75 - q25

        cut_off = iqr * 1.5
```

Продовження ДОДАТКУ В

```
    lower, upper = q25 - cut_off, q75 + cut_off
    outliers = 0

    for x in en_df[i]:
       if x < lower or x > upper:
          en_df = en_df[en_df[i] != x]
          outliers = outliers + 1
    outliers_removed = [x for x in en_df[i] if lower < x < upper]

    # ax3.hist(en_df[i])
    # ax4.boxplot(en_df[i])
    # plt.show()
    # print('Znaydeno vykydiv: %d' % outliers)
    # print('Zalyshylos ryadkiv: %d' % len(outliers_removed))
    # print()

  with open('../data/preprocessed_df.pk', 'wb') as fi:
    # dump your data into the file
    pdump(en_df, fi)

  return en_df
```

main_modeling.py - вфдповідає за навчання моделі та її збереження

```
import pandas as pd
from tensorflow.random import set_seed
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LeakyReLU
import numpy as np
from pickle import load as pload
from pickle import dump as pdump
import matplotlib.pyplot as plt
from preprocess import preprocess
# встановлюємо зерно
seed = 2

# 1. Set `PYTHONHASHSEED` environment variable at a fixed value
from os import environ
from random import seed as rd_seed
environ['PYTHONHASHSEED']=str(seed)
# 2. Set python built-in pseudo-random generator at a fixed value
rd_seed(seed)
# 3. Set numpy pseudo-random generator at a fixed value
np.random.seed(int(seed))
# 4. Set the tensorflow pseudo-random generator at a fixed value
set_seed(seed)

# це запускається один раз,
# отримання попередньо оброблених даних
df = preprocess()
# # для багаторазових запусків можна завантажувати вже оброблені дані,
# # не має сенсу використовувати разом із попередньою командою
# with open('../data/preprocessed_df.pk', 'rb') as fi:
#     df = pload(fi)

# Відокремлення даних
# поділ на вибірку не потрібний, бо модель тестується в іншому місці
```

Продовження ДОДАТКУ В

```
yg = df['y'].copy()
ix = df.drop('y', axis=1).copy()
ix, yg = ix.values, yg.values


def create_model(input_dim, output_dim, seed, alpha, drop_p, neurons):
    model = Sequential()
    model.add(Dense(neurons, activation=LeakyReLU(alpha=alpha), input_shape=(input_dim,)))
    model.add(Dense(neurons, activation=LeakyReLU(alpha=alpha)))
    model.add(Dense(neurons, activation=LeakyReLU(alpha=alpha)))
    model.add(Dense(neurons, activation=LeakyReLU(alpha=alpha)))
    model.add(Dropout(drop_p, seed=seed))
    model.add(Dense(output_dim, activation='sigmoid'))
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    return model


# Задаємо параметри моделі
output_dim = 1
input_dim = ix.shape[1]
batch_size = 50
epochs = 15
neurons = 50
alpha = 0.01
drop_p = 0.3
# Формуємо модель
model_nn_1 = create_model(input_dim, output_dim, seed, alpha, drop_p, neurons)
model_nn_1.summary()
history = model_nn_1.fit(ix, yg,
                         batch_size=batch_size,
                         epochs=epochs,
                         verbose=0)
print("Модель навчено")
# pd.DataFrame(history.history).plot()
# plt.show()
model_nn_1.save('../models/model_nn_1.h5')
```

encode_new.py - відповідає за енкодерінг даних, що поступають із форми для обробки

```
from pandas import read_csv, DataFrame
from numpy import hstack
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from pickle import load as pload


def encode_new(new_data):
    if not new_data:
        print("exception in encode_new.py parameters was not given")
    else:
        new_data = [[int(i) if i[0].isdigit() else i for i in new_data]]

        # Data reading and uniting with parameters of form
        # це потрібно для того, щоб узяти імена колонок і енодери правильно спрацювали
        base_df                                                                    =
read_csv("D:/00KNU/4_kurs/4_2_diplom/diplom_bovsunovska/src/main/python/data/data.csv",     sep=";",
header=0, index_col=False)
```

Продовження ДОДАТКУ В

```
    df = DataFrame(new_data, columns=base_df.columns)
    # df = DataFrame(pd.concat([df, new_data]))

    # encodering
    toLabel = ["col_19", "col_20", "y"]
    toOne = ["col_1", "col_3", "col_4", "col_6", "col_7", "col_9", "col_10", "col_12", "col_14", "col_15",
"col_17"]
    newNames = ["col_2", "col_5", "col_8", "col_11", "col_13", "col_16", "col_18", "col_19", "col_20",
"y"
        , "col_1.1", "col_1.2", "col_1.3", "col_1.4"
        , "col_3.1", "col_3.2", "col_3.3", "col_3.4", "col_3.5"
        , "col_4.1", "col_4.2", "col_4.3", "col_4.4", "col_4.5", "col_4.6", "col_4.7", "col_4.8", "col_4.9",
            "col_4.10"
        , "col_6.1", "col_6.2", "col_6.3", "col_6.4", "col_6.5"
        , "col_7.1", "col_7.2", "col_7.3", "col_7.4", "col_7.5"
        , "col_9.1", "col_9.2"
        , "col_10.1", "col_10.2", "col_10.3"
        , "col_12.1", "col_12.2", "col_12.3", "col_12.4"
        , "col_14.1", "col_14.2", "col_14.3"
        , "col_15.1", "col_15.2", "col_15.3"
        , "col_17.1", "col_17.2", "col_17.3", "col_17.4"]

    # ініціалізуємо енкодери
    encoder_l = LabelEncoder()
    encoder_ohe = OneHotEncoder(sparse=False)
    with
open('D:/00KNU/4_kurs/4_2_diplom/diplom_bovsunovska/src/main/python/encoders/ohe_classes.npy',
'rb') as fi:
        encoder_ohe = pload(fi)

    encoded_categorical_columns = DataFrame(encoder_ohe.transform(df[toOne]))

    for i in toLabel:
        with
open('D:/00KNU/4_kurs/4_2_diplom/diplom_bovsunovska/src/main/python/encoders/label_encoder_classe
s_'+i+'.npy', 'rb') as fi:
            encoder_l = pload(fi)
        df[i] = encoder_l.transform(df[i])

    df = df.drop(toOne, axis=1)
    df = DataFrame(hstack((df, encoded_categorical_columns)))
    df.columns = [newNames]

    result = df.iloc[-1,:]

    # пересуваємо y в кінець для зручності
    tmp_y = df['y'].copy()
    df.pop('y')
    df['y'] = tmp_y

    # повернути список (з y)
    return result
```

Продовження ДОДАТКУ В

main_prediction.py - відповідає за проведення нормалізації вхідних для проведення предікшну даних а також за саме передбачення

```
from sys import argv
import numpy as np
from encode_new import encode_new
from pickle import load as pload
from numpy import log2
from numpy import argmax
from os import environ

environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
from tensorflow.keras.models import load_model
from create_text_file import create_text_file
# "Тестовий Тест Тестович;0960000000;test@test.com;28.05.2022;0000000001;
# A12;45;A34;A41;4576;A62;A71;3;A91;A101;4;A123;27;A143;A152;1;A173;1;A191;A201;user_login"
from warnings import filterwarnings

filterwarnings('ignore')


def main():
    # отримуємо запит із змінними
    tmp_argv = argv[1:].copy()
    tmp_argv = tmp_argv[0].split(";")
    tmp_personal = tmp_argv[0:5]
    tmp_params = tmp_argv[0:-1]
    tmp_login = tmp_argv[-1]

    # проводимо енкодерінг вхідних, тобто переданих даних
    # не забуваємо, що в кінці є фейкове значення у для нормального енкодерингу
    tmp_params_for_encode= tmp_argv[5:-1].copy()
    tmp_params_for_encode.append("1")
    encoded_new = encode_new(tmp_params_for_encode)

    # проводимо нормалізацію необхідних колонок
    with
open('D:/00KNU/4_kurs/4_2_diplom/diplom_bovsunovska/src/main/python/data/normalization_params_np.pk',
            'rb') as fi:
        normalization_params = pload(fi)
    toSt = ["col_2", "col_5", "col_13"]
    en_log_new = encoded_new.copy()
    for i in toSt:
        en_log_new[i] = (log2(en_log_new[i]) - log2(normalization_params[toSt.index(i)][0])) / \
                        (log2(normalization_params[toSt.index(i)][1]) -
log2(normalization_params[toSt.index(i)][2]))
    en_log_new = en_log_new.tolist()
    # прибираэмо у тепер без фіктивного у:
    en_log_new.pop()
    # декодовані, нормалізовані дані для предікшина
    # print(en_log_new)

    # завантажуємо попередньонавчену модель
    model =
load_model('D:/00KNU/4_kurs/4_2_diplom/diplom_bovsunovska/src/main/python/models/model_nn_1.h5')
```

Продовження ДОДАТКУ В

```
    # робимо передбачення імовірності
    res_y = model.predict([en_log_new], )
    # робимо передбачення класу
    threshold = 0.5
    res_y_class = np.where(res_y > threshold, 1, 0)
    result_string = ";"

    if res_y_class[0] == 0:
        res_message = "Ризики у наданні кредиту низькі"
    else:
        res_message = "Ризики у наданні кредиту високі"
    res_percent = str(int(round(res_y[0][0] * 100, 0)))

    result_string += res_percent+"%" + ";" + res_percent + ";" + str(res_y_class[0])
    # print(res_y)
    # print(res_y_class)
    print(result_string)
    # проводимо збереження даних у файл
    inf = [tmp_login, res_message, res_percent+"%"]
    create_text_file(inf, tmp_params)


main()


        create_text_file.py - відповідає за створення xlsx файлу, який користувач має можливість
                              завантажити, іншими словами - формує звітність по заявці
import pandas as pd

inf = ["user_login", "Ризикованість видачі кредиту низька", "99%"]
data = ["Тестовий Тест Тестович", "0960000000", "test@test.com", "28.05.2022", "0000000001", "A12",
"45", "A34", "A41",
        "4576", "A62", "A71", "3", "A91", "A101", "4", "A123", "27", "A143", "A152", "1", "A173", "1",
"A191", "A201"]
def create_text_file(inf, data):
    # чтение по строкам первого файла
    file = open("D:/00KNU/4_kurs/4_2_diplom/diplom_bovsunovska/src/main/resources/resources.txt",
encoding="UTF-8")
    lines = file.readlines()

    # уборка мусора из певрого набора
    for line in lines:
        if line.startswith('#'):
            lines.remove(line)
    lines = [i[:-1] if i.endswith("\n") else i for i in lines]

    # парсинг первого набора в словарь
    lines = [i.split(" = ") for i in lines]
    resources = {i[0].replace("main.form.", ""): i[1] for i in lines}

    # чтение по строкам второго файла
    file = open("D:/00KNU/4_kurs/4_2_diplom/diplom_bovsunovska/src/main/resources/col_values.txt",
encoding="UTF-8")
    lines = file.readlines()

    # уборка мусора из второго набора
```

Продовження ДОДАТКУ В

```
    for line in lines:
        if line.startswith('#'):
            lines.remove(line)
    lines = [i[:-1] if i.endswith("\n") else i for i in lines]

    # парсинг второго набора в словарь
    lines = [i.split(" = ") for i in lines]
    col_values = {i[1]: i[2].replace("\"", "") for i in lines}

    # паковка в датафреймы
    n = len([i for i in resources.keys() if i.startswith("pers")])
    df1 = pd.DataFrame({"Name": list(resources.values())[:n],
                        "Value": data[:n]})
    df3 = pd.DataFrame({"Name": ["Оцінка ризикованості кредиту",
                                 "Імовірність неповернення кредиту позичальником"],
                        "Value": inf[1:]})
    df1 = df1.append(df3)
    df2 = pd.DataFrame({"Name": list(resources.values())[n:],
                        "Value": [col_values[i] if i in col_values else i for i in data[5:]]})

    # запись в файл
    writer = pd.ExcelWriter("D:/00KNU/4_kurs/4_2_diplom/diplom_bovsunovska/src/main/python/docks/results_for_"+inf[0]+".xlsx")
    df1.to_excel(writer, sheet_name="Results", header=False, index=False)
    df2.to_excel(writer, sheet_name="Details", header=False, index=False)
    writer.close()


create_text_file(inf, data)
```